

# Interoperability/Reusability of high level WebAudio components

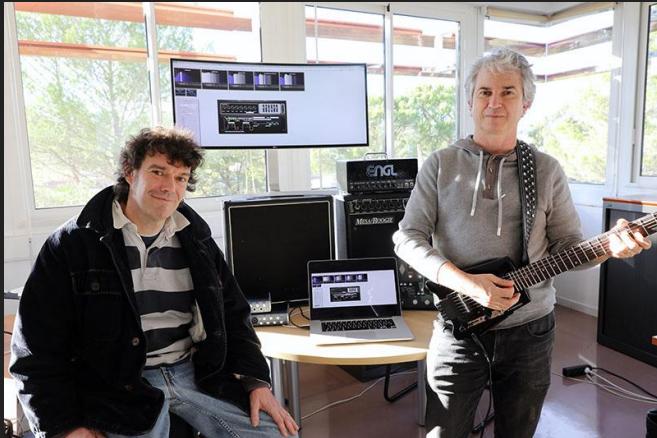
Games on Web W3C Workshop, Seattle 2019

Michel Buffa  
Université Côte d'Azur  
France  
I3S/CNRS/INRIA labs  
[buffa@i3s.unice.fr](mailto:buffa@i3s.unice.fr)  
[@micbuffa](https://twitter.com/micbuffa)



# Who am I?

- Professor / researcher at Université Côte d'Azur, in the South of France
  - member of the WIMMICS research group common to INRIA and I3S lab from CNRS
- National coordinator of the WASABI ANR research project, with WebAudio at its heart,
- W3C Advisory Committee Representative for UCA
- I participate to the WebAudio working group



## Real-Time Emulation of a Marshall JCM 800 Guitar Tube Amplifier, Audio FX Pedals, in a Virtual Pedal Board



# Some ambitious WebAudio examples...

- AudioGraphs of these apps use high-level WebAudio nodes and / or AudioWorklets + WebAssembly.



But... no plugin standard, no “hosts”, no programming model...

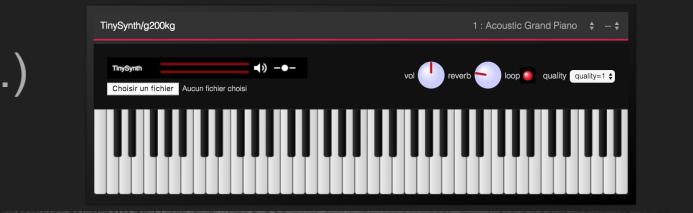
We find some very good JavaScript libraries (i.e. [toneJS](#))

Some open source github repositories (i.e. <https://webaudiodeemos.appspot.com/>)

Some online tools for music synthesis ([genish.js](#) etc.)

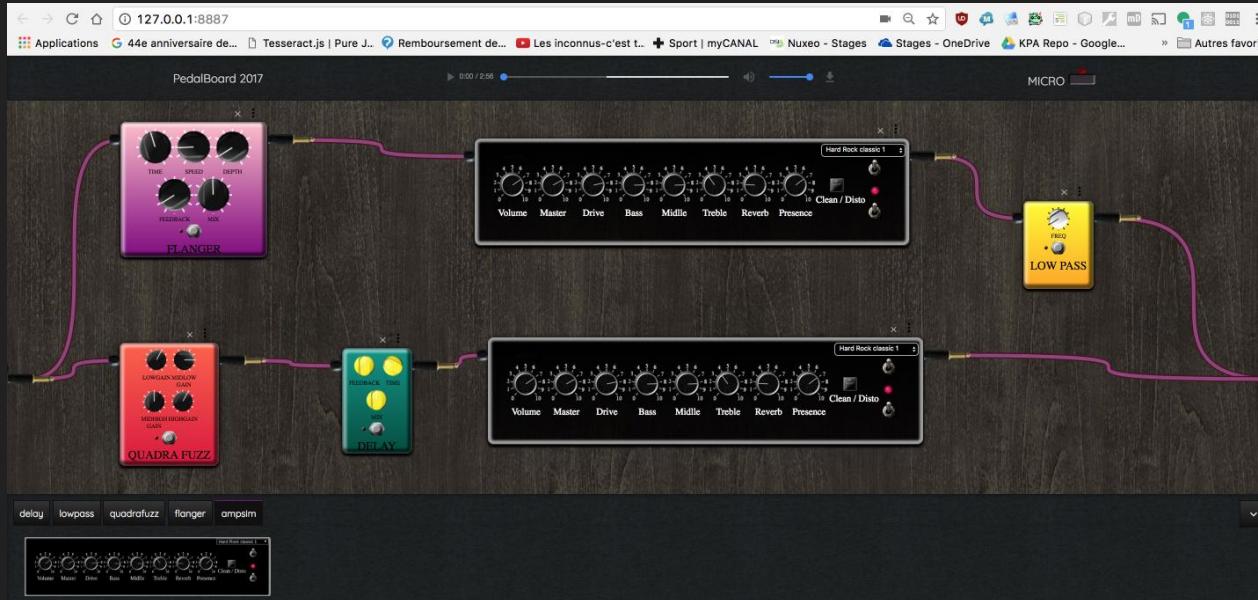
Some DSL for DSP programming ([FAUST](#), etc.)

Some effects and instruments





In early 2018, with some researchers and developers we decided to start working on an open plugin standard for WebAudio



We made a team with different researchers / developers, that share same concerns with different approaches

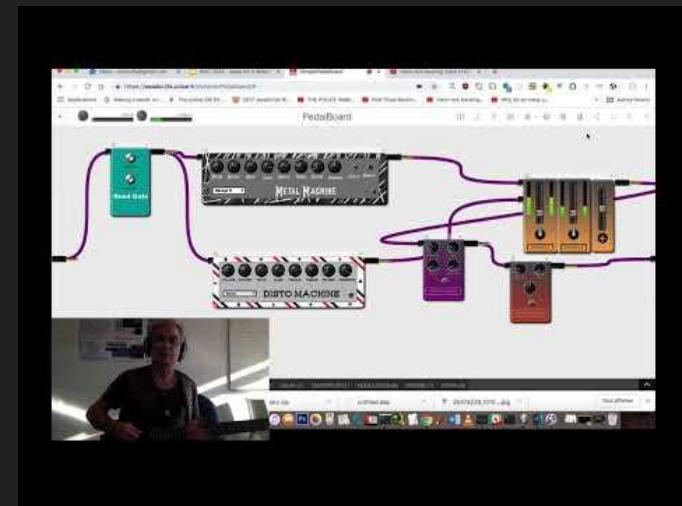
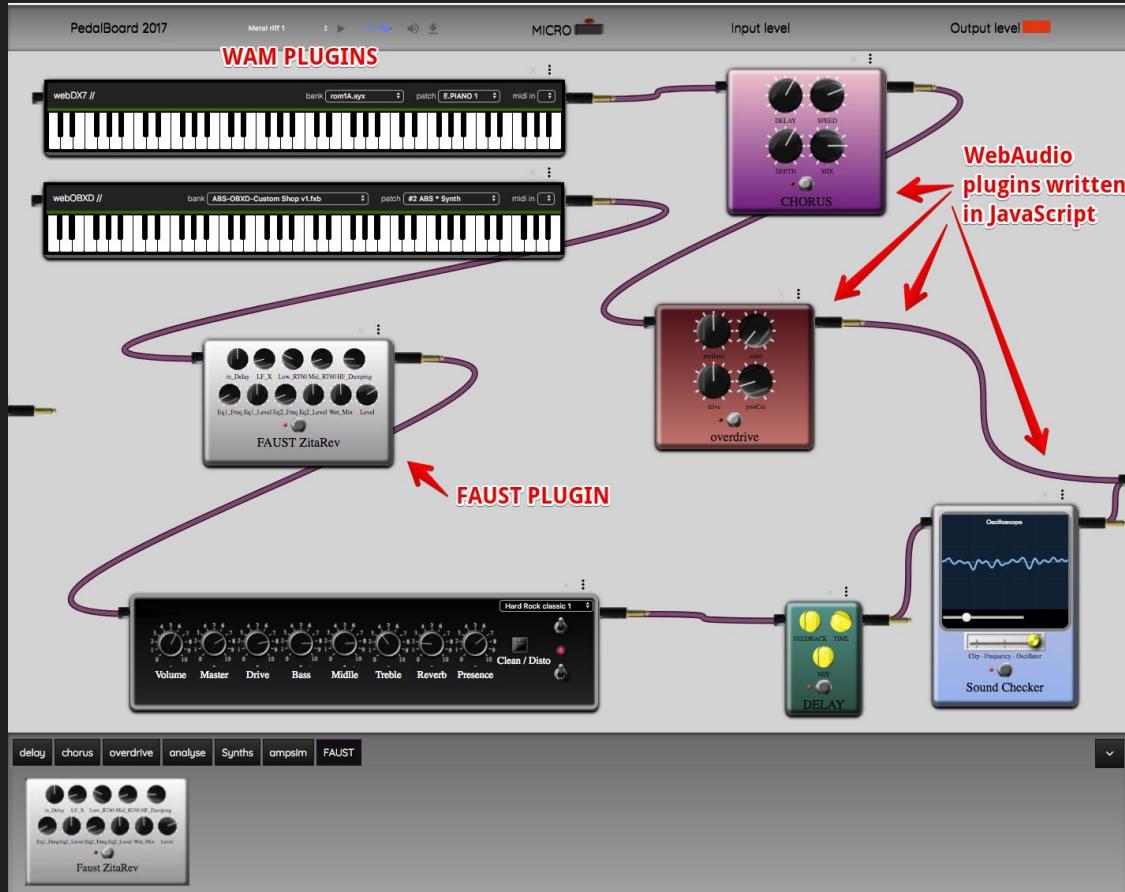
- **1 - Bringing native developers to the Web**
  - a. **Jari Kleimola** (Aalto University Espoo, Southern Finland, now at Webaudiomodules.org),
  - b. **Oli Larkin** (Developer of VirtrualCZ, Endless Series, WDL-OL/iPlug, iPlug2)
- **2 - Bringing low level DSP developers to the Web**
  - a. **Stéphane Letz** (senior researcher at GRAME, Lyon, co-author of the FAUST DSL/compiler)
- **3 - Attract Web Developers / JavaScript audio app developers**
  - a. **Tatsuya Shinyagaito, aka q200kg**, (Audio and WebAudio developer, huge WebAudio contributor, Yokohama, Kanagawa, Japan)
  - b. **Jerôme Lebrun and Michel Buffa** (I3S/INRIA)

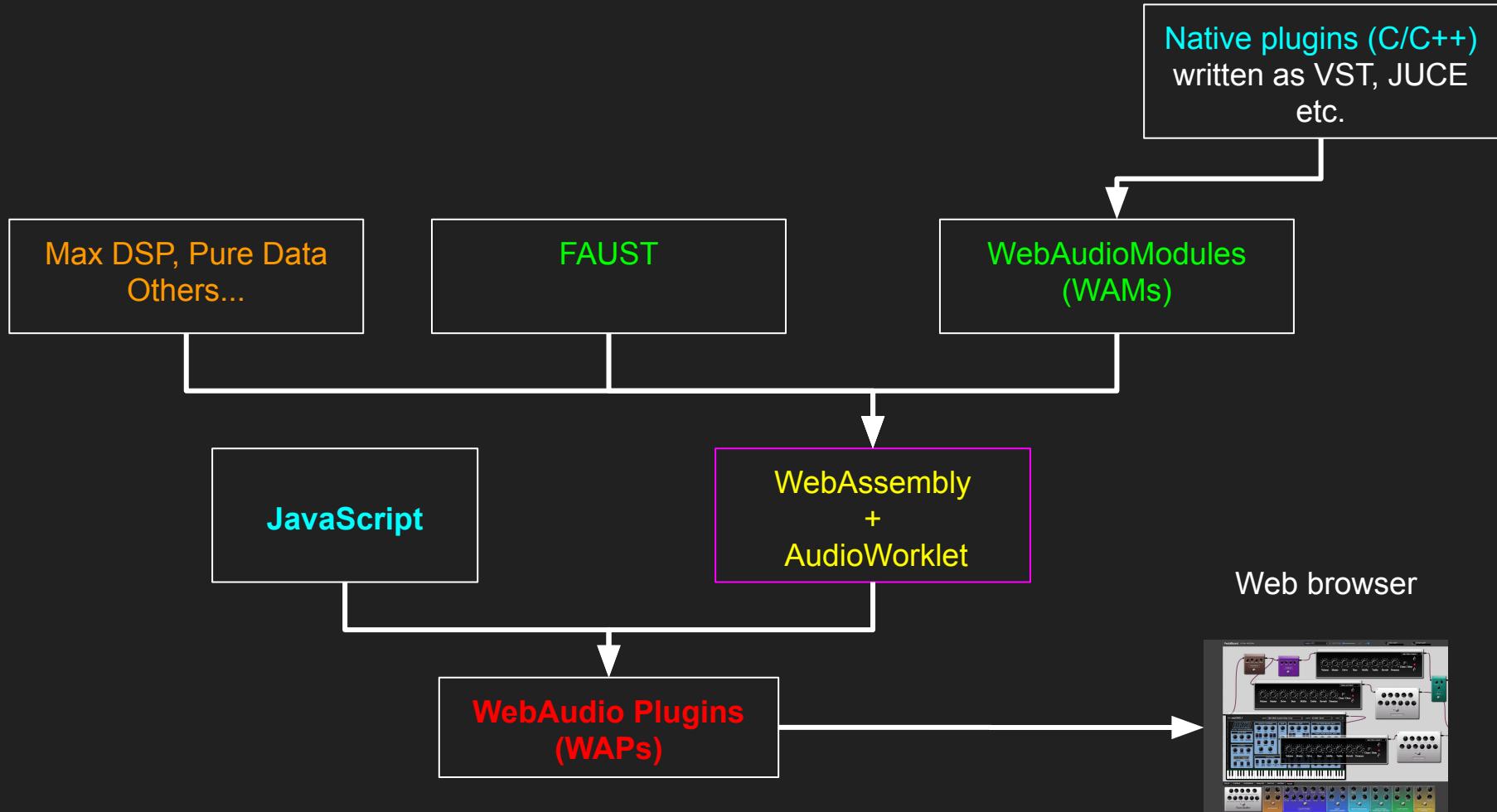
An open standard = API/specification ? Or more... ?

## Be Web-Aware!

- **Use URLs** : support local or remote plugins, audio and midi routing
- **Asynchronous events in the lifecycle**
- **Plugins can be headless or with a GUI**
- **API for the “audio processor part”, as close as possible to AudioNode**
  - i.e use `plugin.connect(gain)` or `gain.disconnect(plugin)`, etc.
- **Propose an API or at least guidelines on how to package and publish plugins on REST repositories**
- **Avoid naming conflicts** (HTML ids, JS names, CSS rules), metadata...

# Let's play with some WAPs (WebAudio Plugins)





Applications JS Sharing Geometry...

Bobrov Dev JS Rocks Panopticlick Scanner et numéri... Evolution by Keiwan (7) Prostatectomi... Machine Learning... Le Deep Learning,... A Pen by Fr0stbyt... Autres favoris

File Add library Share

HTML CSS JavaScript Console Output

Login or Register Blog Help

```
// Once the script has been loaded instanciate the plugin
buildPlugin(className, baseURL);

// will be executed before the onload above...
document.head.appendChild(script);

// instanciate the plugin
function buildPlugin(className, baseURL) {

  var plugin = new window[className](ctx, baseURL);
  console.log(plugin);

  plugin.load().then((node) => {
    // loads and initialize the audio processor part
    // Then use the factory to create the HTML custom elem that holds the GUI
    // The loadGUI method takes care of inserting the link rel=import part,
    // not doing it twice, and associate the node with its GUI.
    if (checkbox.checked) {
      plugin.loadGui().then((elem) => {
        console.log("ADDING PLUGIN");
        // show the GUI of the plugin, the audio part is ready to be used
        document.querySelector("#WAP").appendChild(elem);
        //mediaSource.connect(node);
        //node.connect(ctx.destination);
        // Add node to the chain of plugins
      });
    }
    document.body.querySelector("#WAP").insertAdjacentHTML('afterbegin', '<h2>' + `${className}` +
    try {
      mediaSource.connect(node);
    } catch (error) {
      console.log("this plugin does not use audioworkletnode or compositenode");
      mediaSource.connect(node.getInput(0));
    }

    node.connect(ctx.destination);

    bt_buildIt.addEventListener('click', () => {
      mediaSource.disconnect();
      node.disconnect();
      document.querySelector("#WAP").innerHTML = "";
      document.querySelector("#mocha").innerHTML = "";
    })
  });
}
```

Output

▶ 0:11 / 2:56 ⏪ ⏴ ⏵ ⏹ ⏷

### Paste here the link to your webaudio plugin

Plugin URL (repository where your main.json file is)

<https://wasabi.i3s.unice.fr/WebAudioPluginBank/jordan-S>

GUI



Try different plugins click on links to fill the form field, then press the "build" button:

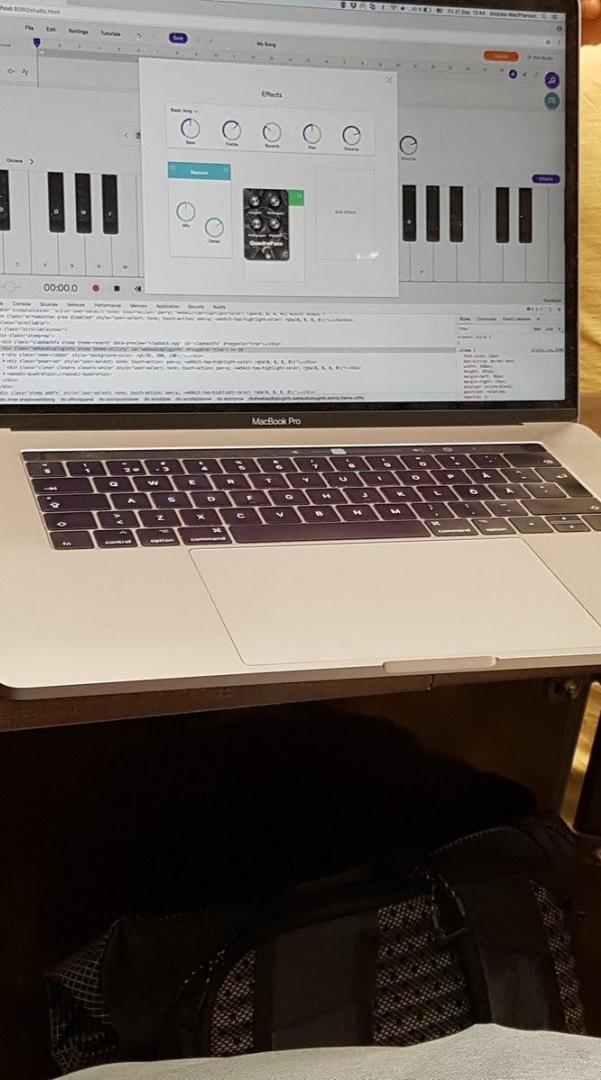
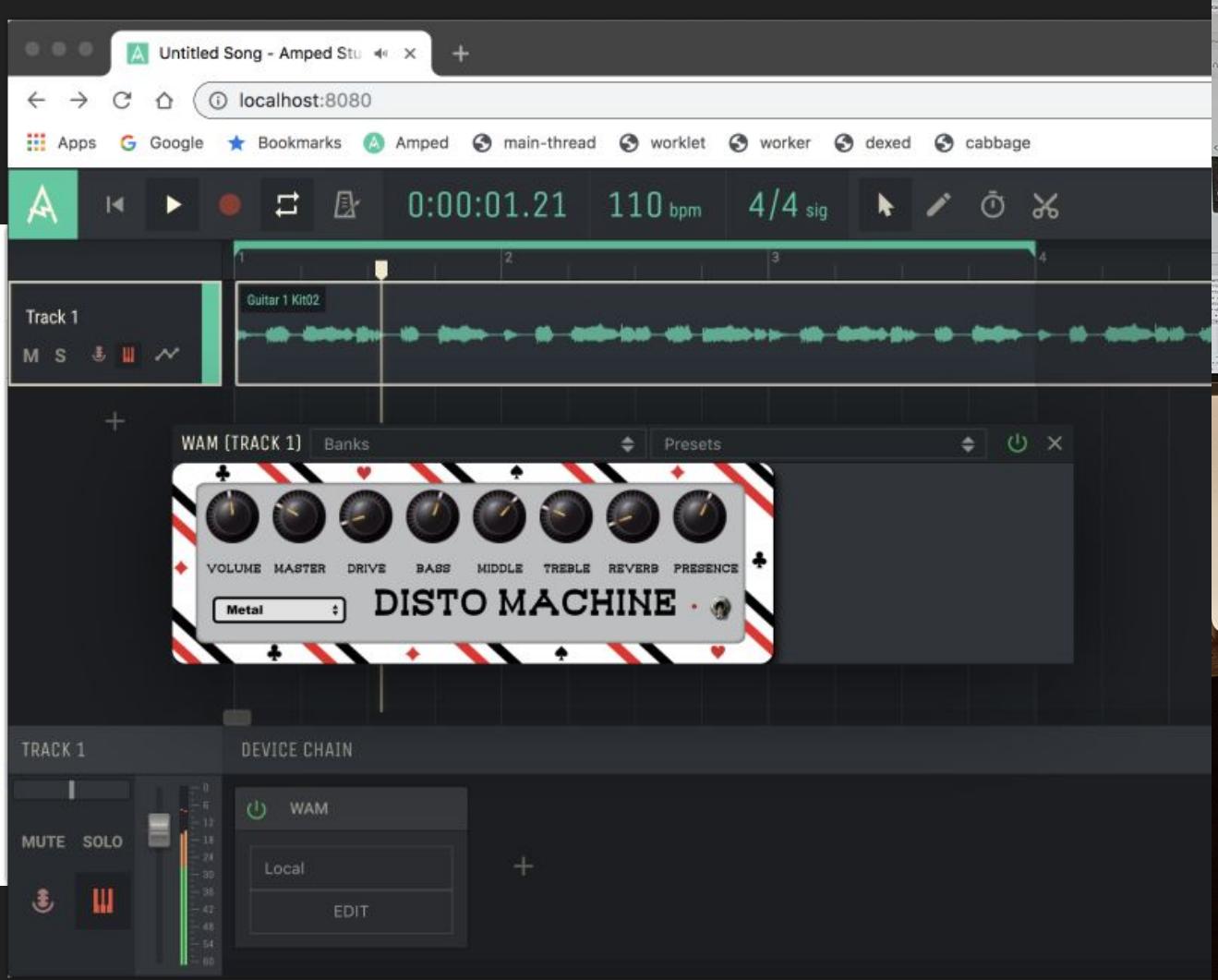
- Metal Machine amp sim
- Acoustic guitar amp sim
- PingPongDelay3
- ZitaRev
- Blipper
- Flanger

### WasabiDistoMachine



micbuffa

9 May



# Low level DSP automatized -> WASM

The screenshot shows two software windows side-by-side. The left window is the 'Faust Editor' showing a DSP code snippet for a 'Blipper8' effect. The right window is the 'WAP GUI2 editor' showing a graphical user interface (GUI) for the same effect.

**Faust Editor (Left):**

- Run** button
- Share** button
- Poly Voices**: Mono dropdown
- Buffer Size**: 1024 dropdown
- Save Options** checkboxes:
  - Use AudioWorklet
  - Save DSP Code (checked)
  - Save Params State
  - Save DSP Cache
  - Real-time Compile (checked)
  - Popup UI
- Plot** section with Mode (Offline), Samples (1024), Sample Rate (48000), FFT Size (256), and Draw Spectrogram checkbox.

```
blipper8 .dsp Examples *
13.59.153.54:8080/fausteditorweb/dist/
1 block_on(i) = par(i, inputs(fx), _);
2 block_off(fx) = par(i, inputs(fx), _);
3
4 bypass_fx(fx) = par(i, inputs(fx), _): ((block_on(fx):fx), block_off(fx)):> par(i, outputs(fx), _);
5
6 basepitch = hslider("BasePitch [unit:semitones] [OWL:PARAMETER_A][style:knob]", 60, 24, 96, 0.1) : si.smooth(ba.tau2pole(0.01));
7 pitchmod = hslider("PitchMod [unit:semitones] [OWL:PARAMETER_B][style:knob]", 24, -64, 64, 1) : si.smooth(ba.tau2pole(0.005));
8 //attack = hslider("Attack [unit:ms] [OWL:PARAMETER_C][style:knob]", 2, 2, 1000, 1) : *(0.001) : max(1.0/float(ma.SR));
9 release = hslider("Release [unit:ms] [OWL:PARAMETER_C][style:knob]", 20, 2, 100, 1) : *(0.001) : max(1.0/float(ma.SR));
10 attack = 0.005;
11 mix = hslider("Mix[OWL:PARAMETER_D][style:knob]", 0.5, 0, 1, 0.01) : si.smooth(ba.tau2pole(0.005));
12
13 blipper(l, r) = l, r <= *(1-mix), mono2stereo := _;
14
15 with {
16     mono2stereo += pc2 * mix <: _;
17     pc2 = an.amp_follower_ud(attack, release) <: (ba.midikey2hz(basepitch + (pitchmod * _)): os.triangle), _ : *
18 };
19 process = bypass_fx(blipper);
```

**WAP GUI2 Editor (Right):**

- Diagram**, **Plot**, **DSP**, and **GUI Builder** tabs. The **GUI Builder** tab is active.
- Welcome on the WAP GUI2 editor ! click on MENU to start !**
- GUI Components** (BasePitch, Mix, PitchMod, bypass, Release) arranged on a brick wall background.
- Properties** for the GUI components:
  - Name:
  - Background color:
  - Image opacity:
  - Width:  px
  - Height:  px
  - Radius:  px
  - Background images:

# Online testers (individual plugin, repository)

**Plugin Tester**

Paste here the link to your webaudio plugin

URL: <https://wasabi.i3s.unice.fr/WebAudioPluginBank/repository.json> Load it Test it

100% passes: 12 failures: 5 duration: 0.05s

**Metadata**

- ✓ plugin should have a JSON getMetadata() method
- ✓ the getMetadata() function should return a json object

**Descriptor**

- ✓ plugin should have a JSON getDescriptor() method
- ✓ getDescriptor() function should return a json object

**Param getter**

\* plugin should have a getParam(key) method

```
AssertionError: expected undefined to exist at Context.<anonymous> (test.html:135:54)
```

\* the getParam() function should not be empty

```
AssertionError: .empty was passed non-string primitive undefined at Context.<anonymous> (test.html:138:61)
```

**WasabiQuadraFuzz**

Enter the URL of a Web-Audio plugin repository (path to the json file which describe it) and press the button to start discovering/loading plugin thumbnails.

Click on a thumbnail to create an instance of the plugin Test its compatibility with the API. Here the input default value is set to 0.5. You can also test it with https://webaudiodom.org/repository.json You can try the WAM repository <https://webaudiodom.org/repository.json>

100% passes: 14 failures: 0 duration: 0.02s

**Metadata**

- ✓ plugin should have a JS02 getMetadata() function

**Descriptor**

- ✓ plugin should have a JS02 getDescriptor() function

**Param getter**

- ✓ plugin should have a getParam() method

**Param setter**

- ✓ plugin should have a setParam() method

**Plugin Tester**

Click to fill with the URL of :

WasabiPingPongDelay - FaustZitaRev - LarkinBlipper -  
WasabiMinilogue - CleanMachine - QuadraFuzz

**Node Part**

URL : node.URL

```
"https://wasabi.i3s.unice.fr/WebAudioPluginBank/WASABI/PingPongDelay3,"
```

Descriptor : node.getDescriptor()

```
{"mix":0,"minValue":0,"maxValue":1,"defaultValue":0.5,"time":0.5,"feedback":0.5}
```

Node input number node.numberOfInputs : 1

Params at this time node.getState() (promise)

```
{"mix":0.5,"time":0.5,"feedback":0.5,"status":"disabled"}
```

Metadata node.getMetadata() (promise)

```
{"documentation":...,"name":"PingPongDelay","thumbnail":"WasabiPingPongDelay3"}
```

**Shows declared params, inputs, etc.**

# Conclusion : Where are we today?

SDK for JS developers

FAUST scripts and IDE that compile .dsp files to WAPs, embedded GUIs builder, publish to remote servers

WebAudioModules C/C++ toolchain for native audio developers (and WAMs are WAPs)

Multiple examples of hosts that load plugins dynamically using URIs

Tools: plugin validator, repository validator, GUI editor

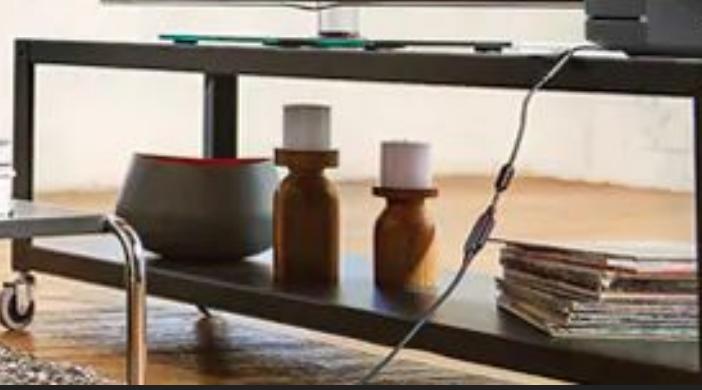
[Check the GitHub!](#)

[Check the pedalboard demo!](#)





16



# Guitar Hero / Rocksmith for the Web?

Yes, this is pos

- Good low
- Pitch de

Still the laten  
addressed on

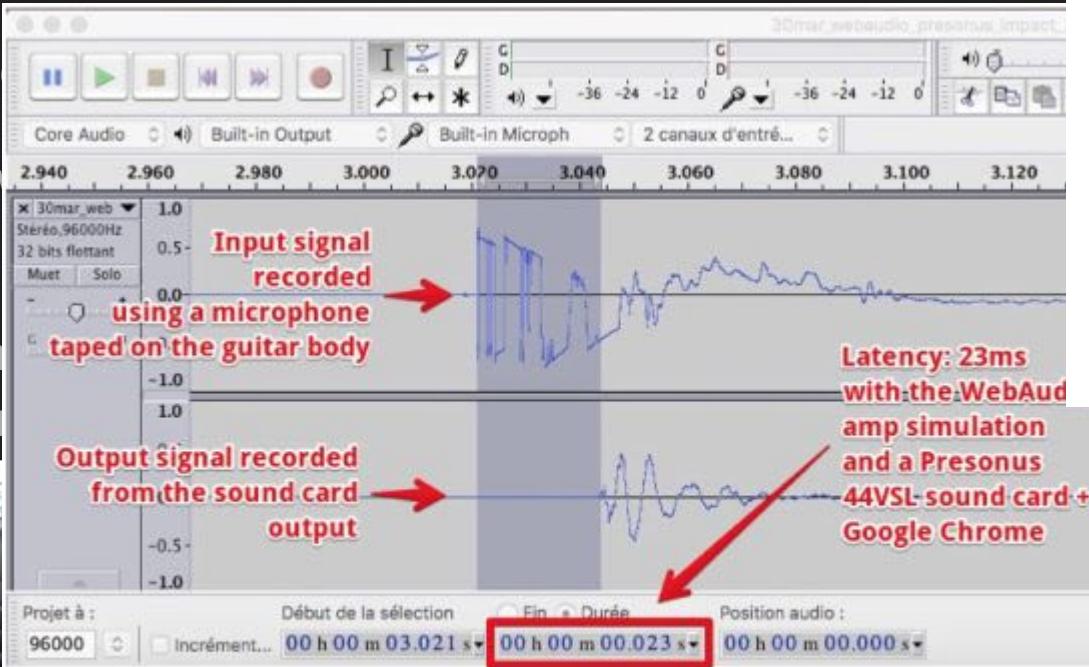


Figure 7: Example of measure in Audacity, here the WebAudio amp sim with Google Chrome and a Presonus 44VSL sound card.

