Interoperability/Reusability of high level WebAudio components

Michel Buffa

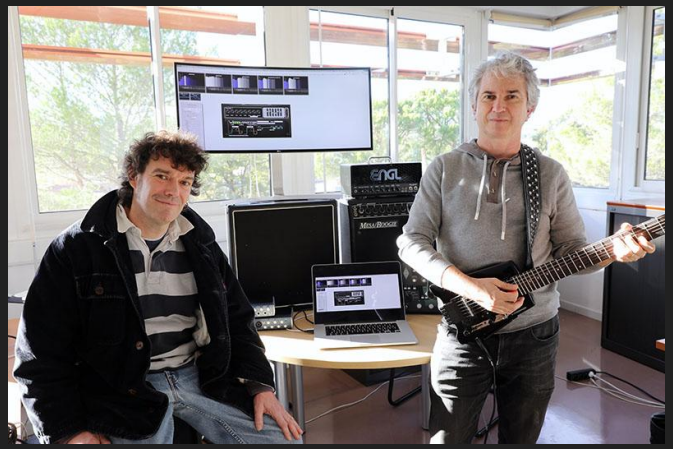
Interoperability/Reusability of high level WebAudio components

Games on Web W3C Workshop, Seattle 2019

Michel Buffa
Université Côte d'Azur
France
I3S/CNRS/INRIA labs
buffa@i3s.unice.fr
@micbuffa

Who am I?

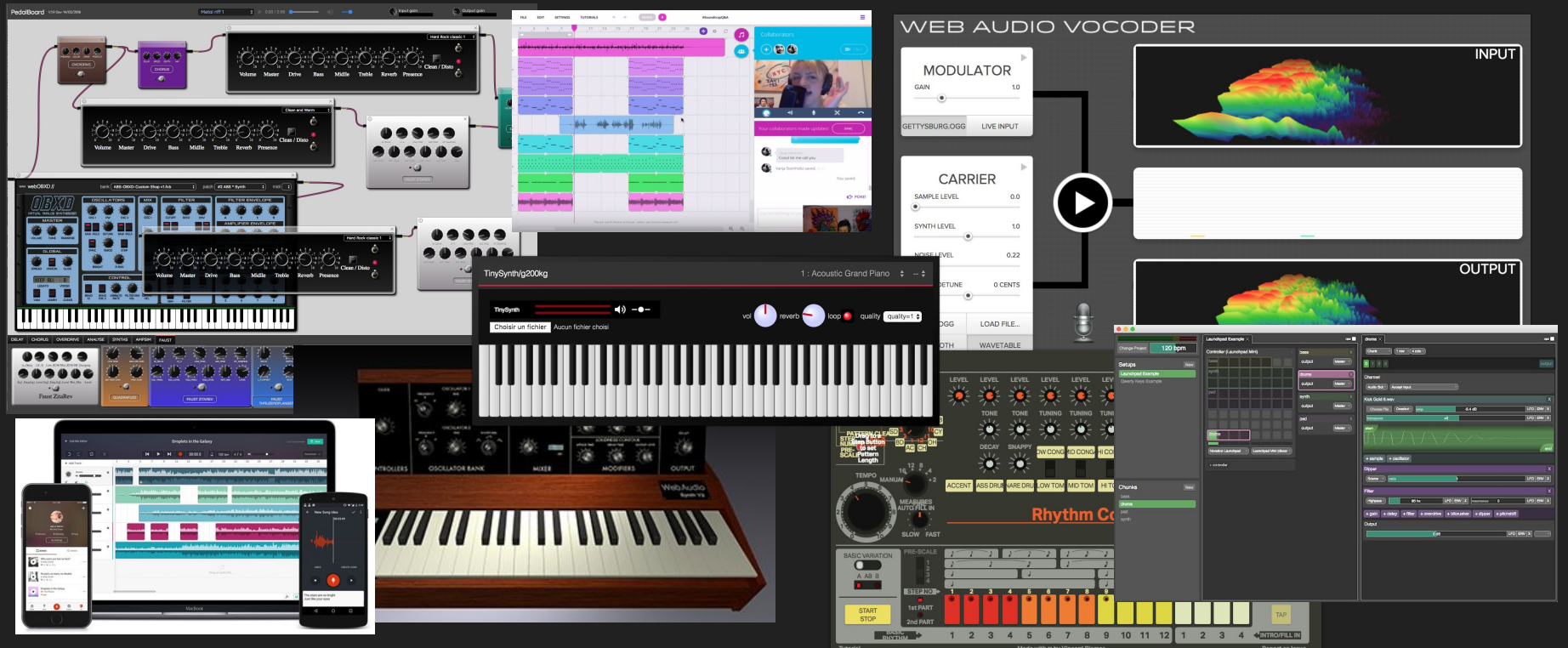
- Professor / researcher at Université Côte d’Azur, in the South of France
  - member of the WIMMICS research group common to INRIA and I3S lab from CNRS
- National coordinator of the WASABI ANR research project, with WebAudio at its heart,
- W3C Advisory Committee Representative for UCA
- I participate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




Real-Time Emulation of a Marshall JCM 800 Guitar Tube Amplifier, Audio FX Pedals, in a Virtual Pedal Board
Michel Buffa, Jerome Lebrun
Université Côte d'Azur
CNRS, INRIA
(buffa, lebrun)@i3s.unice.fr
WebAudio / WebMidi APIs
proposal for an open plugin standard (ala VST/VSTi for the Web)
Targets: music schools, composers,...
Guitar amp schematics
Tube guitar amp simulator/designer, final amps packaged as plugins
Host Webapp for assembling amps/effects and instrument plugins
Plugins can be:
JavaScript classes
WebComponents (GUI)
WebAssembly (binary)
C/C++ compiled
FAUST compiled
Support for local / remote plugins
Integrated Midi support
Synths by webaudiomodules.org
FAUST DSL: http://faust.grame.fr
ANR WASABI
http://wasabihome.i3s.unice.fr

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...

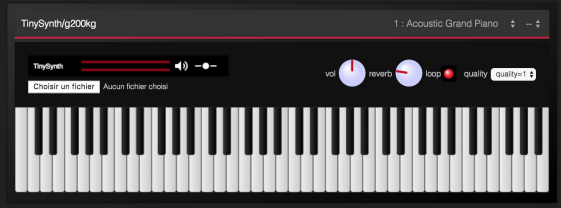
We find some very good JavaScript libraries (i.e. toneJS)

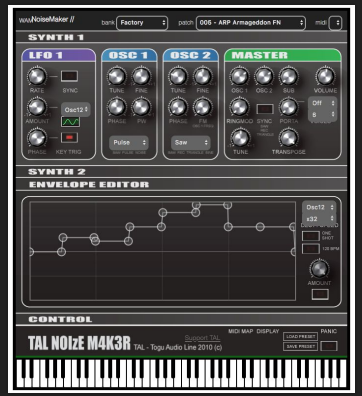
Some open source github repositories (i.e. https://webaudiodemos.appspot.com/)

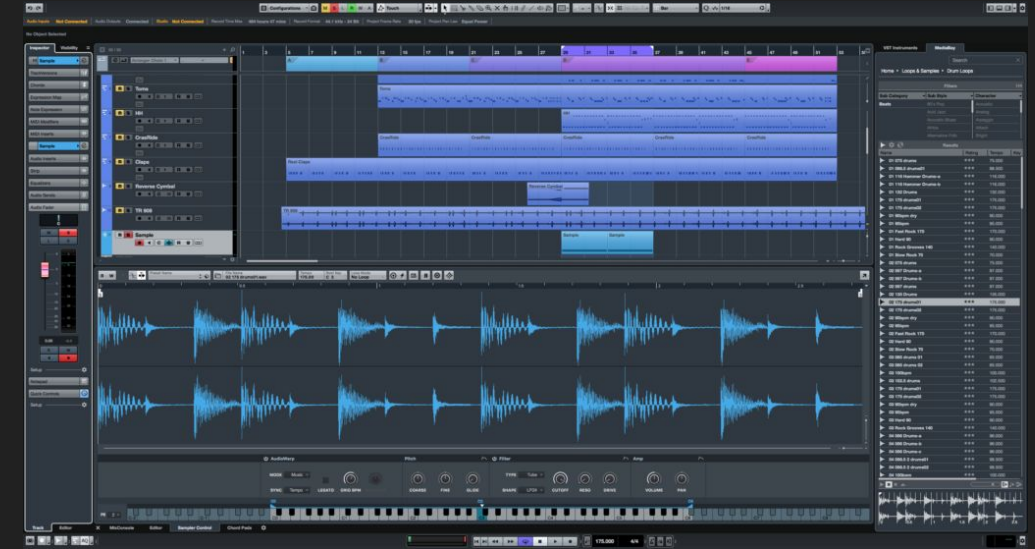
Some online tools for music synthesis (genish.js etc.)

Some DSL for DSP programming (FAUST, etc.)

Some effects and instruments

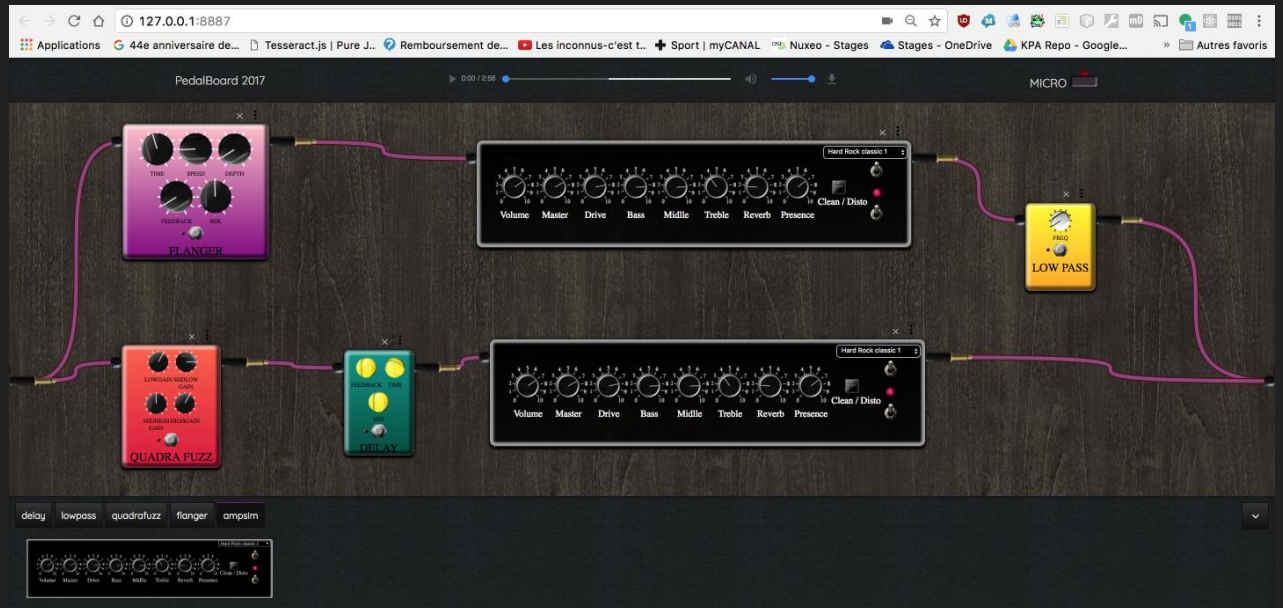







Vibe Phaser
Tri-Knob Fuzz
Spring Reverb

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 g200kg**, (Audio and WebAudio developer, huge WebAudio contributor, Yokohama, Kanagawa, Japan)
  b. **Jerôme Lebrun and Michel Buffa** (I3S/INRIA)

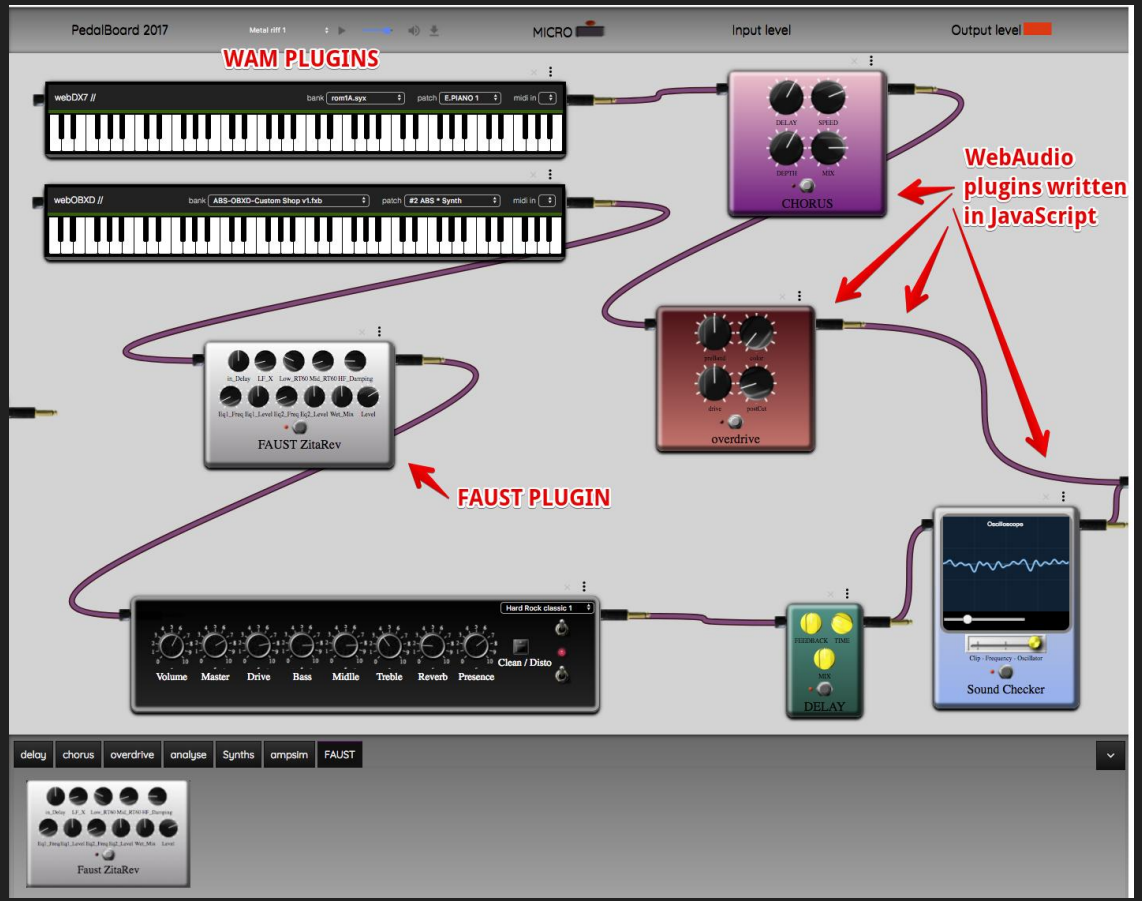
An open standard = API/specification ? Or more… ?

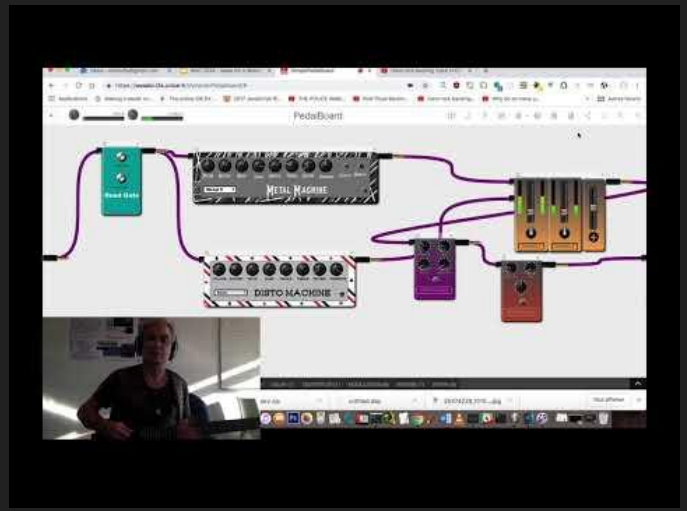
Be Web-Aware!

- **Use URIs : 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)





Native plugins (C/C++) written as VST, JUCE etc.

Max DSP, Pure Data Others...

FAUST

WebAudioModules (WAMs)

JavaScript

WebAssembly + AudioWorklet

WebAudio Plugins (WAPs)

Web browser

https://jsbin.com/gahunof/edit?js,output

Applications | 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 plugn, 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

Run with JS | Auto-run JS

0:11 / 2:56

Paste here the link to your webaudio plugin

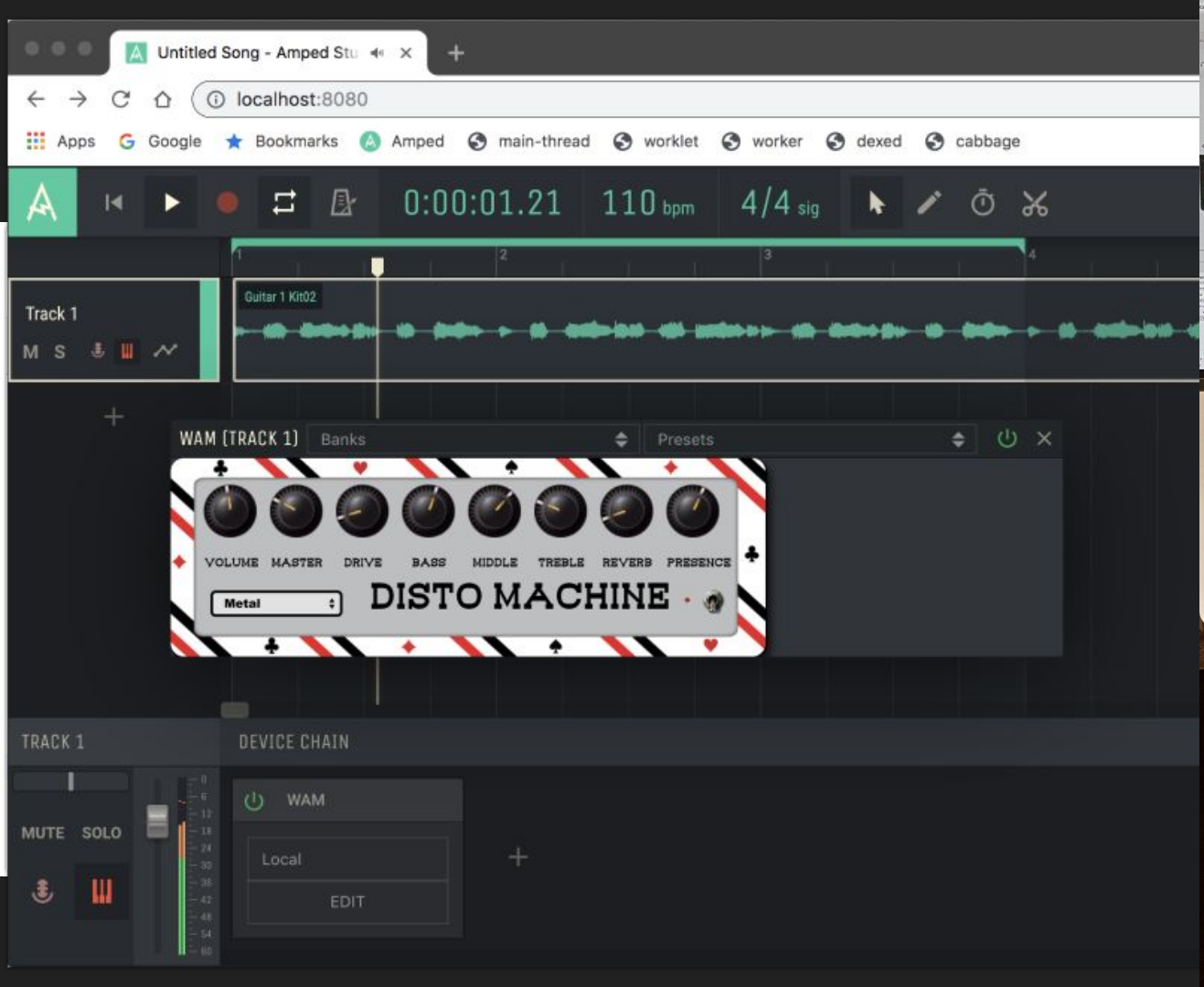
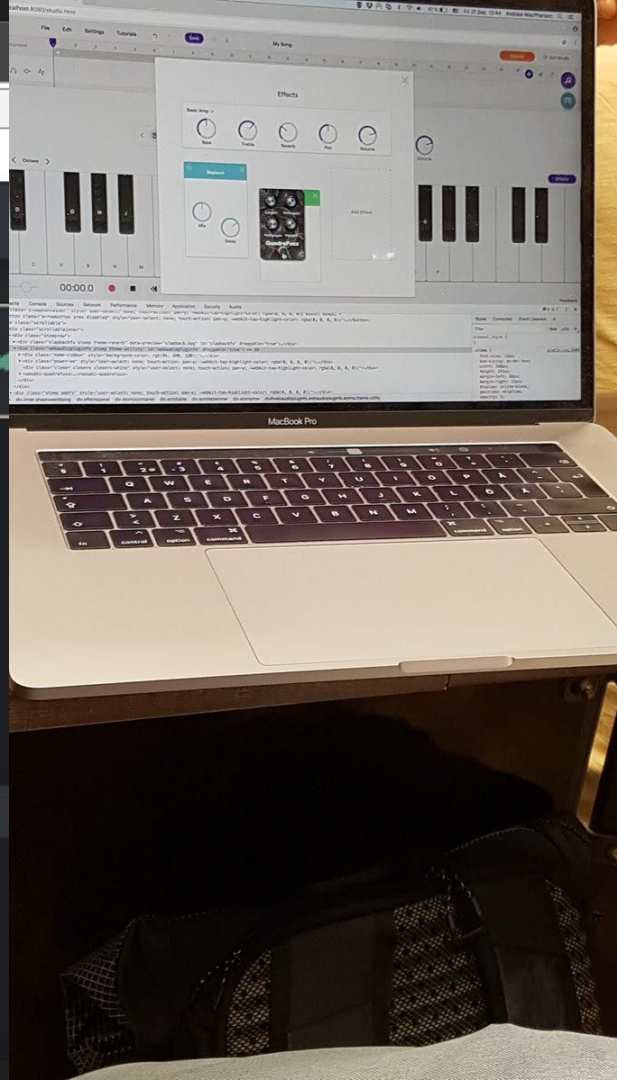
Plugin URL (repository where your main.json file is)
https://wasabi.i3s.unice.fr/WebAudioPluginBank/jordan-S
GUI
build it

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

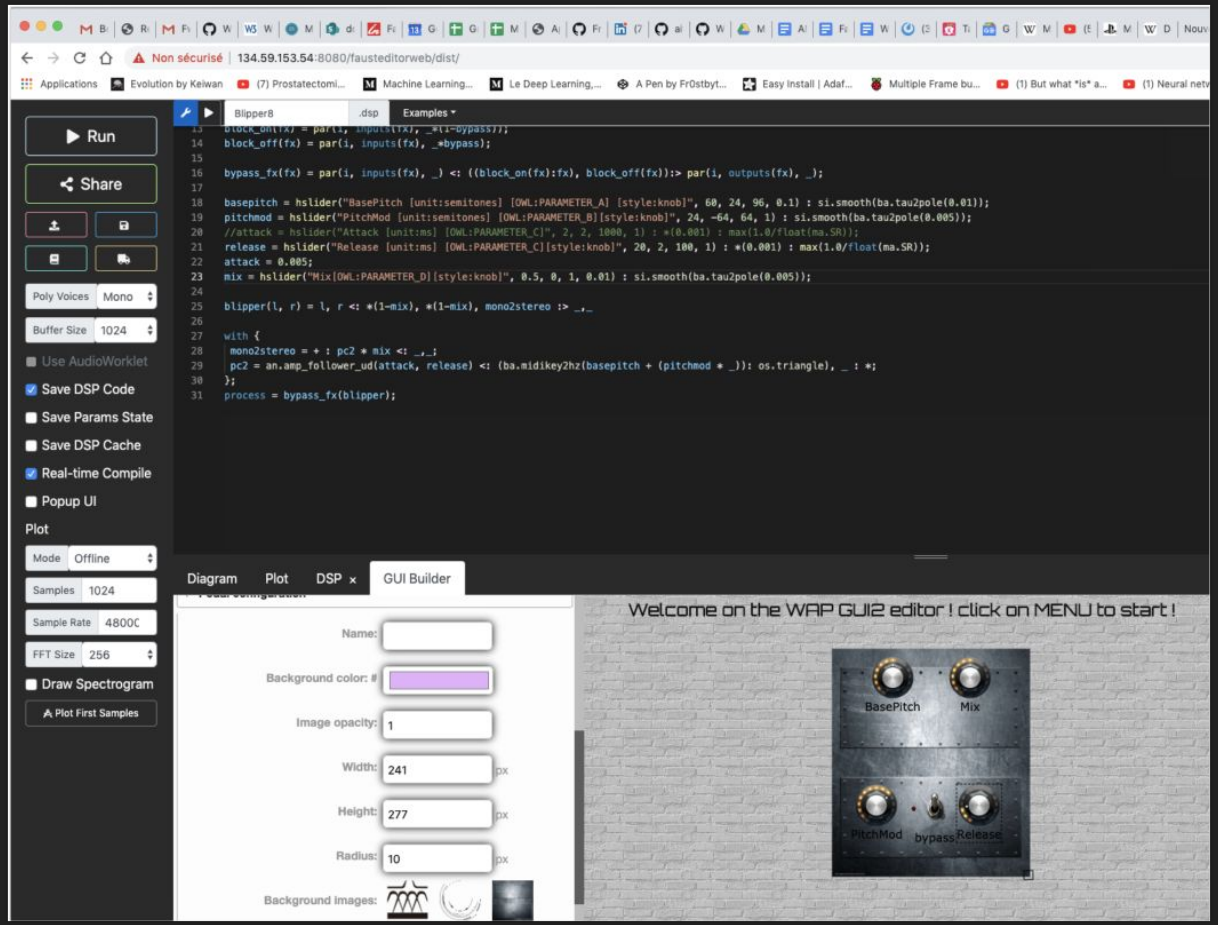
- Metal Machine amp sim
- Acoustic guitar amp sim
- PingPongDelay3
- ZitaRev
- Blipper
- Flanger

WasabiDistoMachine

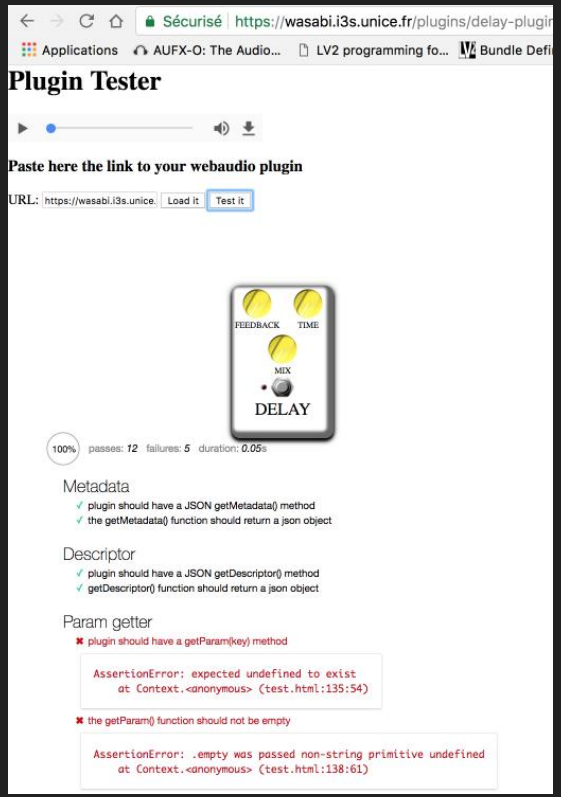
micbuffa
9 May

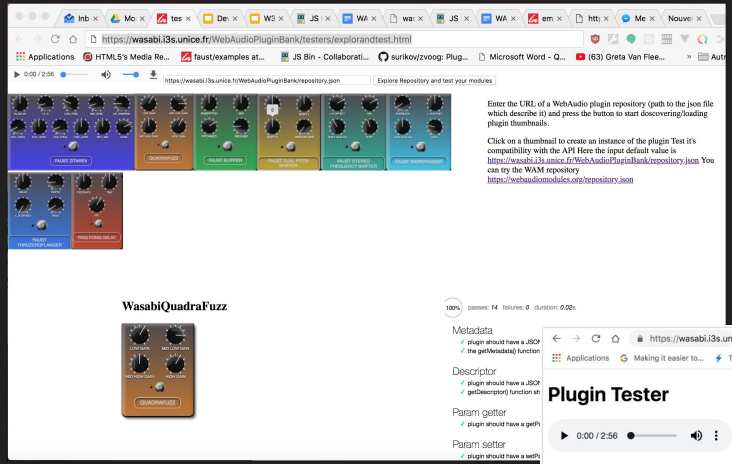

Untitled Song - Amped Stu
localhost:8080
Apps
Google
Bookmarks
Amped
main-thread
worklet
worker
dexed
cabbage
0:00:01.21
110 bpm
4/4 sig
Track 1
M S
Guitar 1 Kit02
WAM (TRACK 1)
Banks
Presets
VOLUME
MASTER
DRIVE
BASS
MIDDLE
TREBLE
REVERB
PRESENCE
Metal
DISTO MACHINE
TRACK 1
DEVICE CHAIN
MUTE
SOLO
WAM
Local
EDIT

MacBook Pro

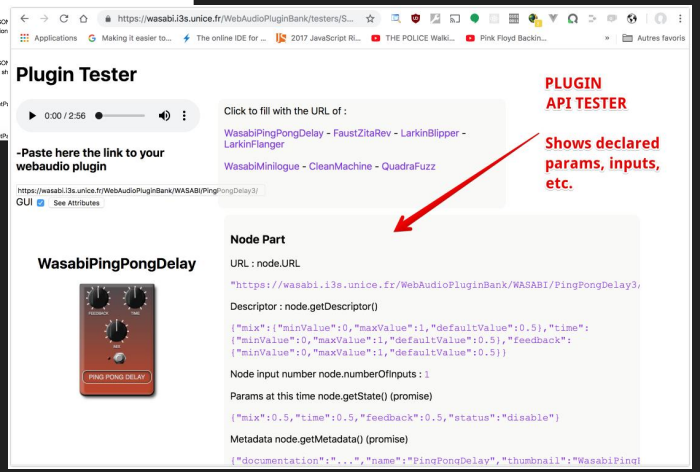
Low level DSP automatized -> WASM



Online testers (individual plugin, repository)







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!


LEARN A SONG
SESSION MODE
NONSTOP PLAY
LESSONS
GUITAR ARCADE
MULTIPLAYER
TONE DESIGNER
SHOP
UPLAY
Rocksmith

Guitar Hero / Rocksmith for the Web?

Yes, this is p

- Good lo
- Pitch de

Still the laten
addressed o

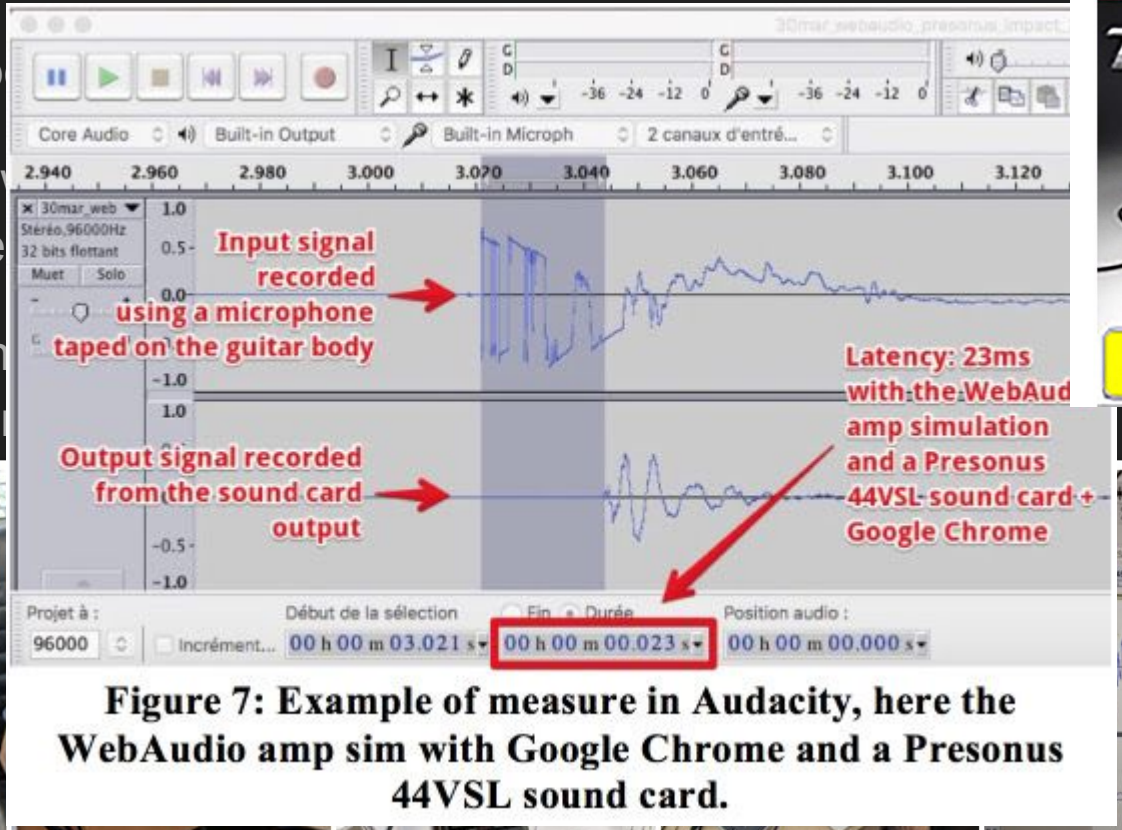


Figure 7: Example of measure in Audacity, here the WebAudio amp sim with Google Chrome and a Presonus 44VSL sound card.