

# Web Audio

# Introduction

# Technologies web - *introduction*

## *Internet*

Ensemble de protocoles de transfert de données (TCP, IP, ...) pour l'interconnection de réseaux

**1972** - Première démonstration officielle d'*ARPANET*  
*envoi d'un message entre UCLA et Stanford*

**1983** - *ARPANET* est officiellement renommé *Internet*

## *World Wide Web*

Système de partage d'informations hypertexte fonctionnant sur Internet

**1989-1992** - Développement au CERN par T. Berners Lee et *Robert Caillau (logiciels et protocoles)*

**1993** - Le CERN ouvre les technologies au domaine public

# Basics

**HTTP**

*HyperText Transfert Protocol*

**URL**

*Uniform Resource Locator*

**HTML**

*HyperText Markup Language*



# Languages

*html / css / javascript*

```
(function hide() {  
    $list.forEach(($el, index) => {  
        $el.classList.remove('hide');  
  
        if (index === counter) {  
            $el.classList.add('hide');  
        }  
    });  
  
    counter = (counter += 1) % $list.length;  
    timeoutId = setTimeout(hide, 1000);  
}());
```

<http://127.0.0.1:3000>

idiot exo?

# Web as a Creative Platform & Web Audio API

# web as creative platform

## *ubiquity*

almost every device implements web standards

## *interactive multimedia*

HTML5/CSS, Web GL, Canvas, Web Audio API,  
DeviceMotion/Orientation, Geolocation

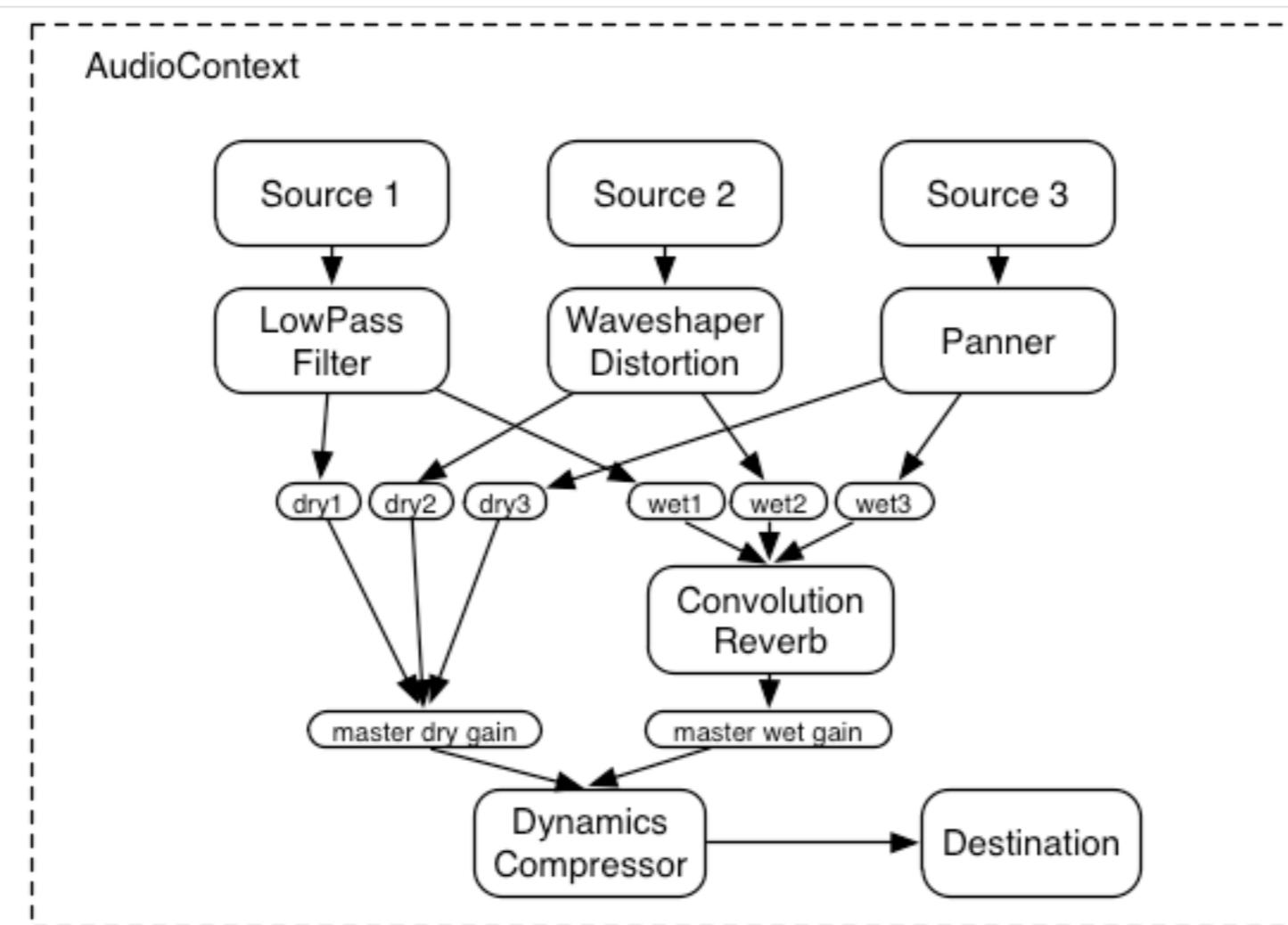
## *networking*

HTTP, WebSockets, WebRTC

## *rapid prototyping & interoperability* (using Node.js)

very rapid development / deployment cycles

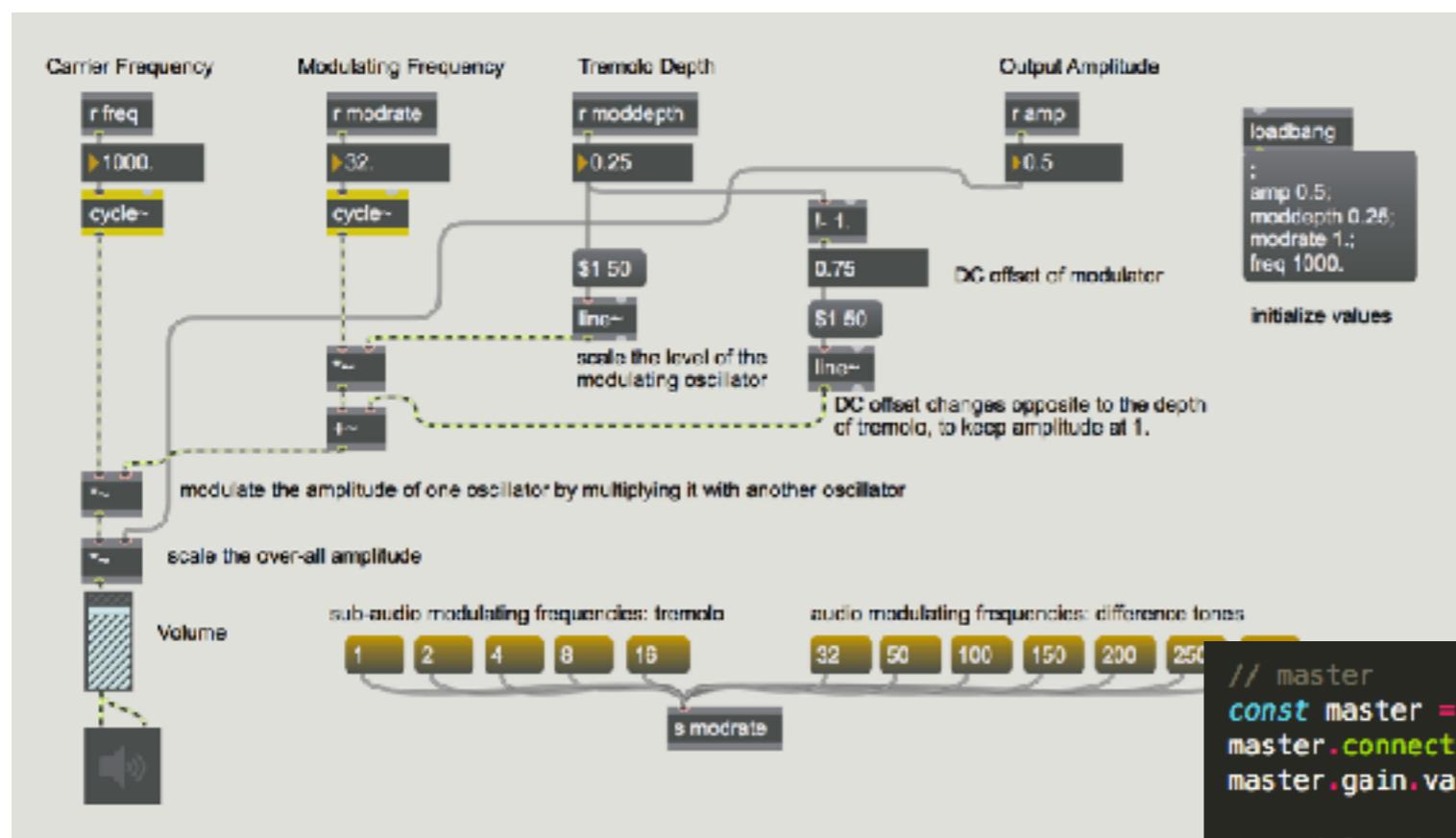
# WebAudio



*specification*

<https://webaudio.github.io/web-audio-api/>

# WebAudio



```
// master
const master = audioContext.createGain();
master.connect(audioContext.destination);
master.gain.value = 0; // default to muted

// modulated amplitude
const amplitude = audioContext.createGain();
amplitude.connect(master);
amplitude.gain.value = 1 - defaultDepth;
amplitude.gain.setValueAtTime(1 - defaultDepth, audioContext.currentTime);

// carrier
const carrier = audioContext.createOscillator();
carrier.connect(amplitude);
carrier.frequency.value = 1000;

// modulation
const depth = audioContext.createGain();
depth.connect(amplitude.gain);
depth.gain.value = defaultDepth
depth.gain.setValueAtTime(defaultDepth, audioContext.currentTime);

const mod = audioContext.createOscillator();
mod.frequency.value = 1;
mod.connect(depth);

carrier.start(audioContext.currentTime);
mod.start(audioContext.currentTime);
```

exo sine + env

# TP 1 - Basics

`./TP/1-04-trigger-sine-with-env`

# Examples

# Librairies

# waves-ui

<http://wavesjs.github.io/waves-ui/examples/layer-waveform.html>

<http://wavesjs.github.io/waves-ui/examples/layer-axis.html>

<http://wavesjs.github.io/waves-ui/examples/layer-segment.html>

<http://wavesjs.github.io/waves-ui/examples/states-zoom.html>

# waves-audio

<https://rawgit.com/wavesjs/waves-audio/master/examples/player-engine/index.html>

<https://rawgit.com/wavesjs/waves-audio/master/examples/granular-engine/index.html>

# waves-lfo

<https://cdn.rawgit.com/wavesjs/waves-lfo/master/examples/sink-vu-meter-display/index.html>

<https://cdn.rawgit.com/wavesjs/waves-lfo/master/examples/sink-waveform-display/index.html>

<https://cdn.rawgit.com/wavesjs/waves-lfo/master/examples/operator-segmenter/index.html>

<https://cdn.rawgit.com/wavesjs/waves-lfo/master/examples/mosaicking/index.html>

# démonstrateurs

<https://wave.ircam.fr/demo/phoenix-1901/>

<https://wave.ircam.fr/demo/bachothèque/>

<https://wave.ircam.fr/demo/leroux-voirex/>

# Artistic projects

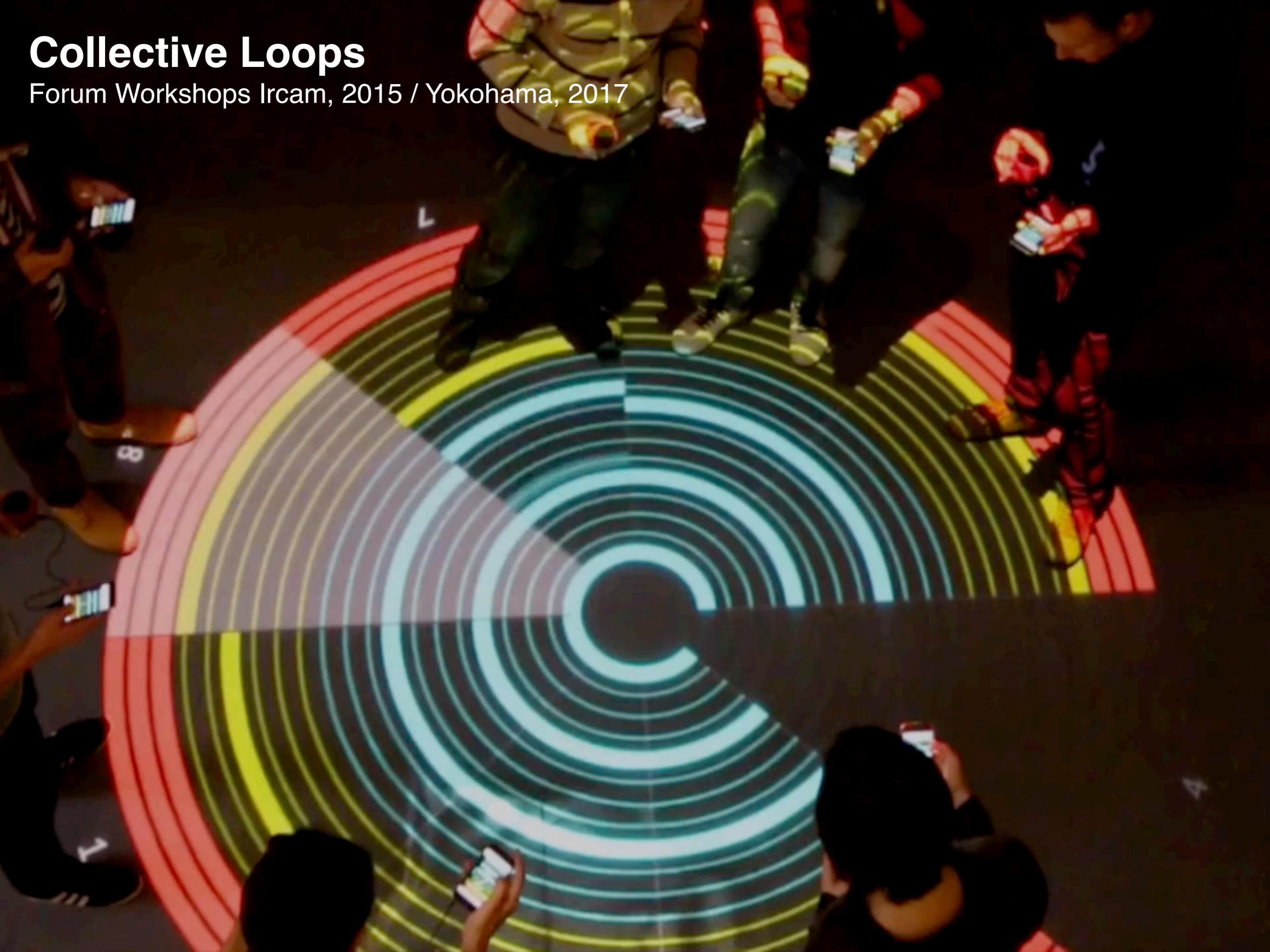
# Collective Sound Checks

2014



# Collective Loops

Forum Workshops Ircam, 2015 / Yokohama, 2017



# 88 Fingers

*AudioMostly 2017*

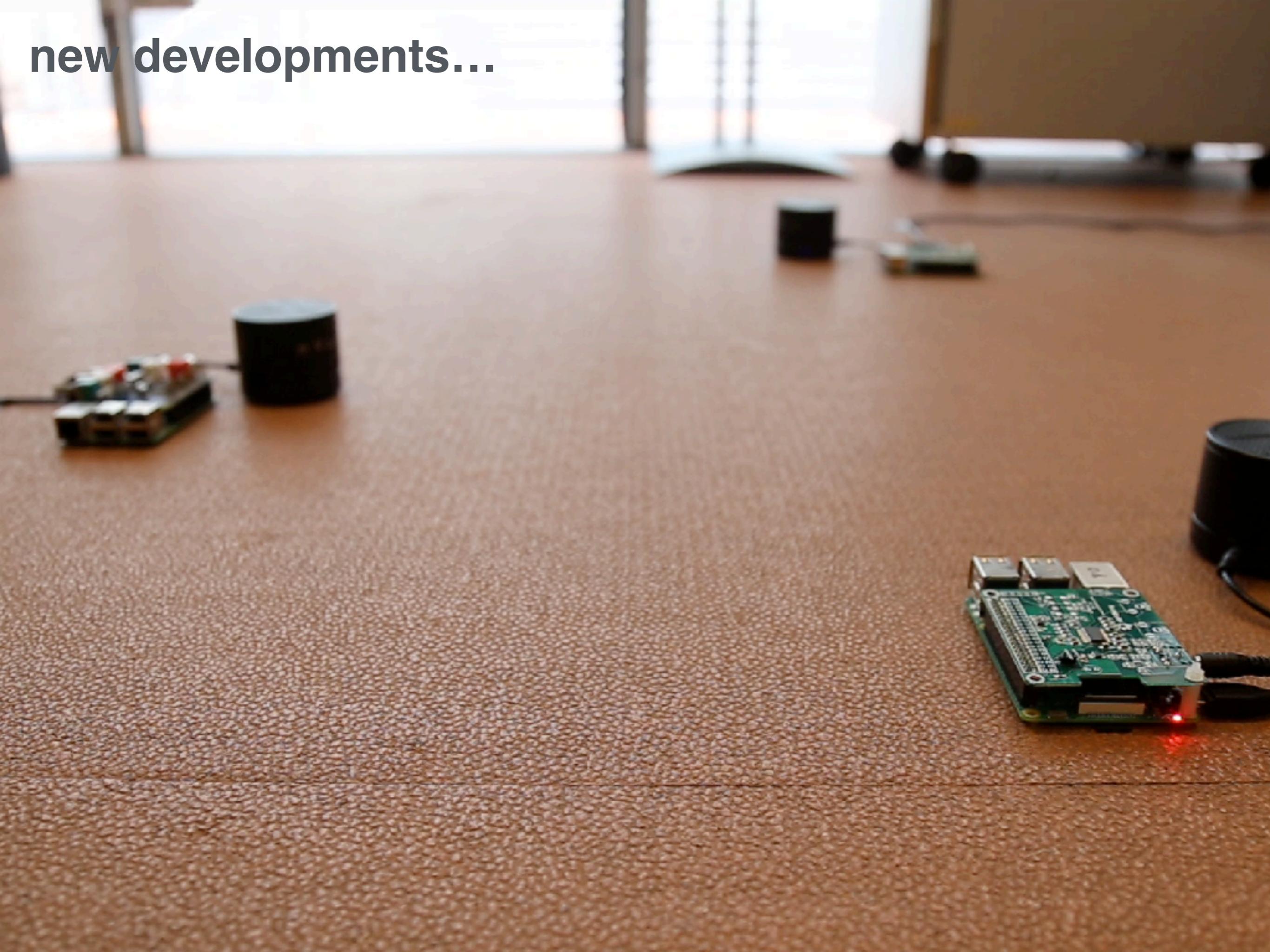


# GrainField

Audio Mostly, London, 2017

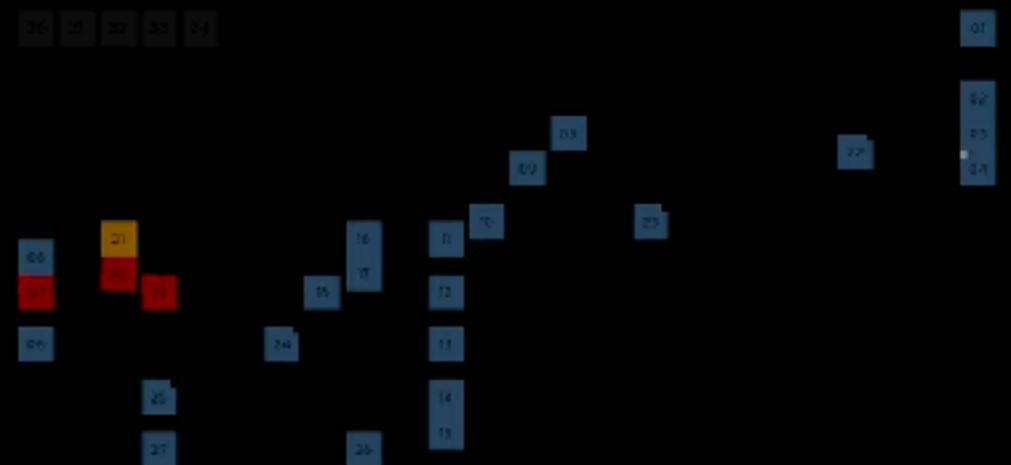


# new developments...



# *Biotope* - Jean-Luc Hervé

2019 - exposition *La fabrique du vivant* - CGP



# Industry

[Save to Patch Pcl](#)[Save to PC](#)[Store to Circuit](#)Synth 1

## Synth 2

Macros

Oscillators

Filter

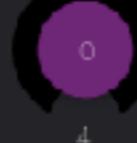
Envelopes

LFO

FX

Settings

## Macro Editor



## Macro 1

No Destination

No Destination

No Destination

No Destination

Start: 0  
End: 127Start: 0  
End: 127Start: 0  
End: 127Start: 0  
End: 127

Depth

Depth

Depth

Depth

## Modulation Matrix

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 ⓘ

Source

Direct



Direct



Depth



Destination

Osc 1/2 Pitch



Osc 1/2 Pitch



Patch: Initial Patch



Genre: None



Category: None



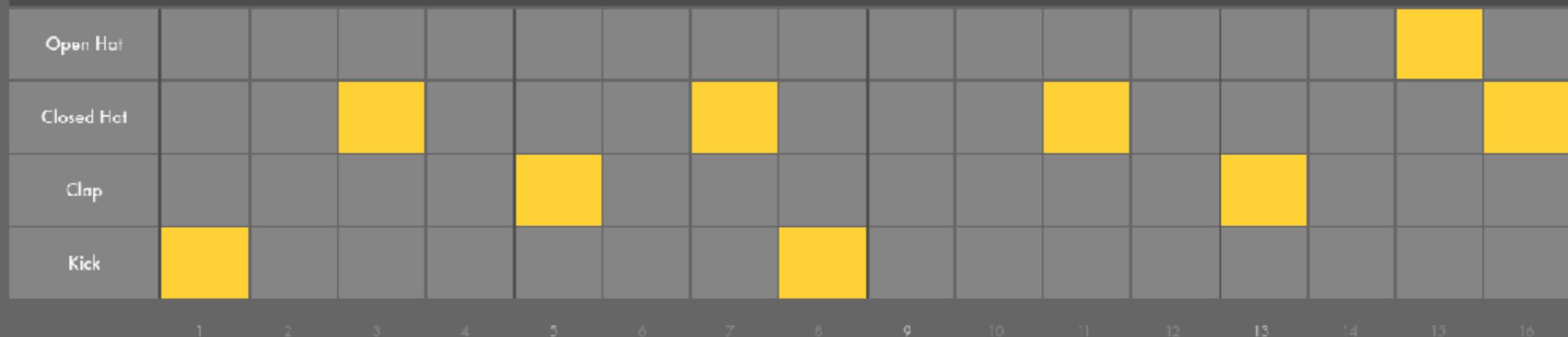
The screenshot shows a DAW interface with the following details:

- Timeline Top Bar:** Displays time 0:00:00.85, tempo 100 bpm, and time signature 4/4 sig.
- Tracks:** The tracks listed on the left are Bass, Drums, Tablas, Synth ARP, Track 5, and Synth ARP. Each track has Mute (M), Solo (S), Automation (A), and Automation Off (U) buttons.
- Sounds:** The Bass, Drums, Tablas, and Synth ARP tracks show waveform clips. The Synth ARP track displays a step sequencer pattern labeled "Synth ARP MR 2".
- Device Chain:** A floating window titled "DEVICE CHAIN" contains a VST rack with the following components:
  - VOLT (selected): Wide Baffles
  - Reverb: Bright Hall
  - Delay
  - Dub SpringEach component has knobs for Size, Damp, Mix, Time, Feedback, Mix, Damp, Cross, and Offset.
- Sound Library:** A sidebar titled "SOUND LIBRARY" lists 18 sound entries with columns for Name, BPM, and Key. Most sounds have a preview icon (Speaker).



Clear Export to Live

Tempo: 100 bpm



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

< Previous

Next: Backbeats >

# MISC

# *Design Patterns*

In **software engineering**, a **software design pattern** is a general, reusable solution to a commonly occurring problem within a given context in **software design**. It is not a finished design that can be transformed directly into **source** or **machine code**. Rather, it is a description or template for how to solve a problem that can be used in many different situations.

[https://en.wikipedia.org/wiki/Software\\_design\\_pattern](https://en.wikipedia.org/wiki/Software_design_pattern)

**Gang of Four:** Design Patterns: *Elements of Reusable Object-Oriented Software* (1994)

**History:** Christopher Alexander, *Notes on the Synthesis of Form* (1964)

→ creating / inventing / designing a *form* adapted to a *context*

# *Design Patterns - 2 examples*

## Singleton

Why?

Implementation

## EventEmitter / PubSub

Why?

Implementation

# TP-1 - Simple Granular Engine (+ Step Sequencer?)

```
git clone https://github.com/b-ma/ATIAM-2019-TP1-boilerplate.git tp-1  
cd tp-1  
npm install  
npm run dev
```

## Resources

<https://www.html5rocks.com/en/tutorials/audio/scheduling/>

[https://wac.ircam.fr/pdf/wac15\\_submission\\_19.pdf](https://wac.ircam.fr/pdf/wac15_submission_19.pdf)

# TP-2 - Distributed Step Sequencer

```
git clone https://github.com/b-ma/ATIAM-2019-TP2-boilerplate.git tp-2
```

```
cd tp-2
```

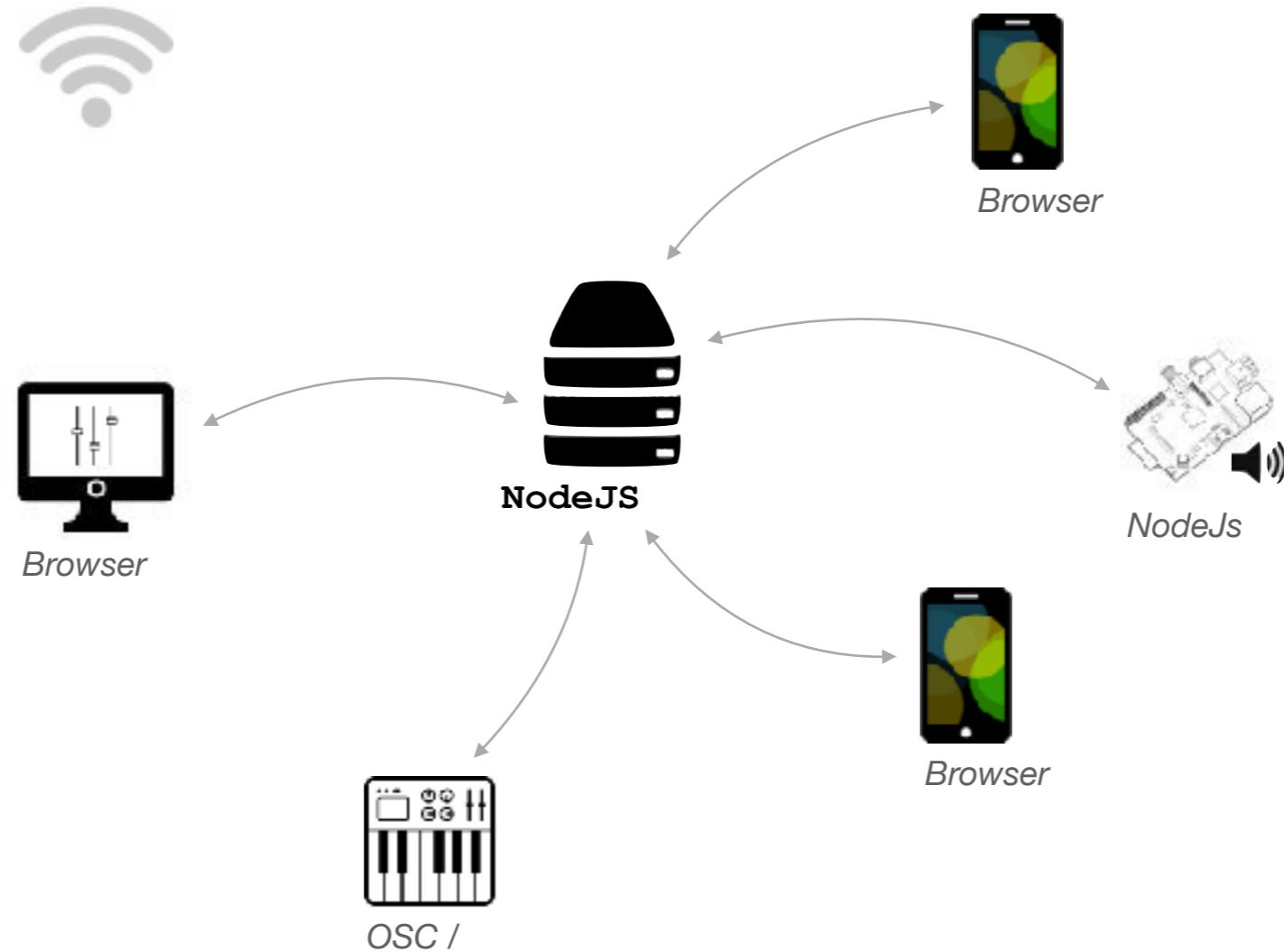
```
npm install
```

```
npm run watch
```

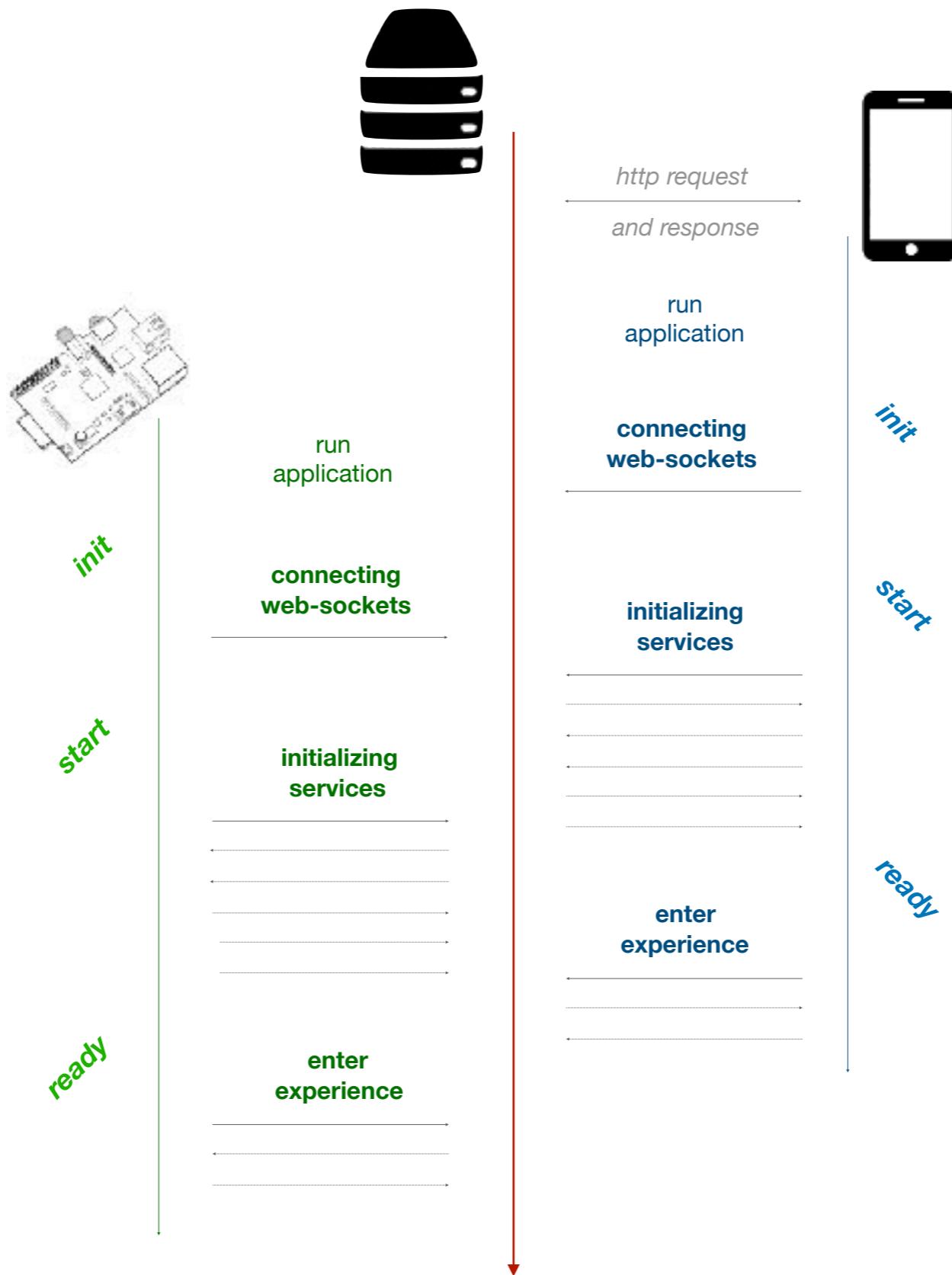
## Resources

<https://hal.archives-ouvertes.fr/hal-02387783>

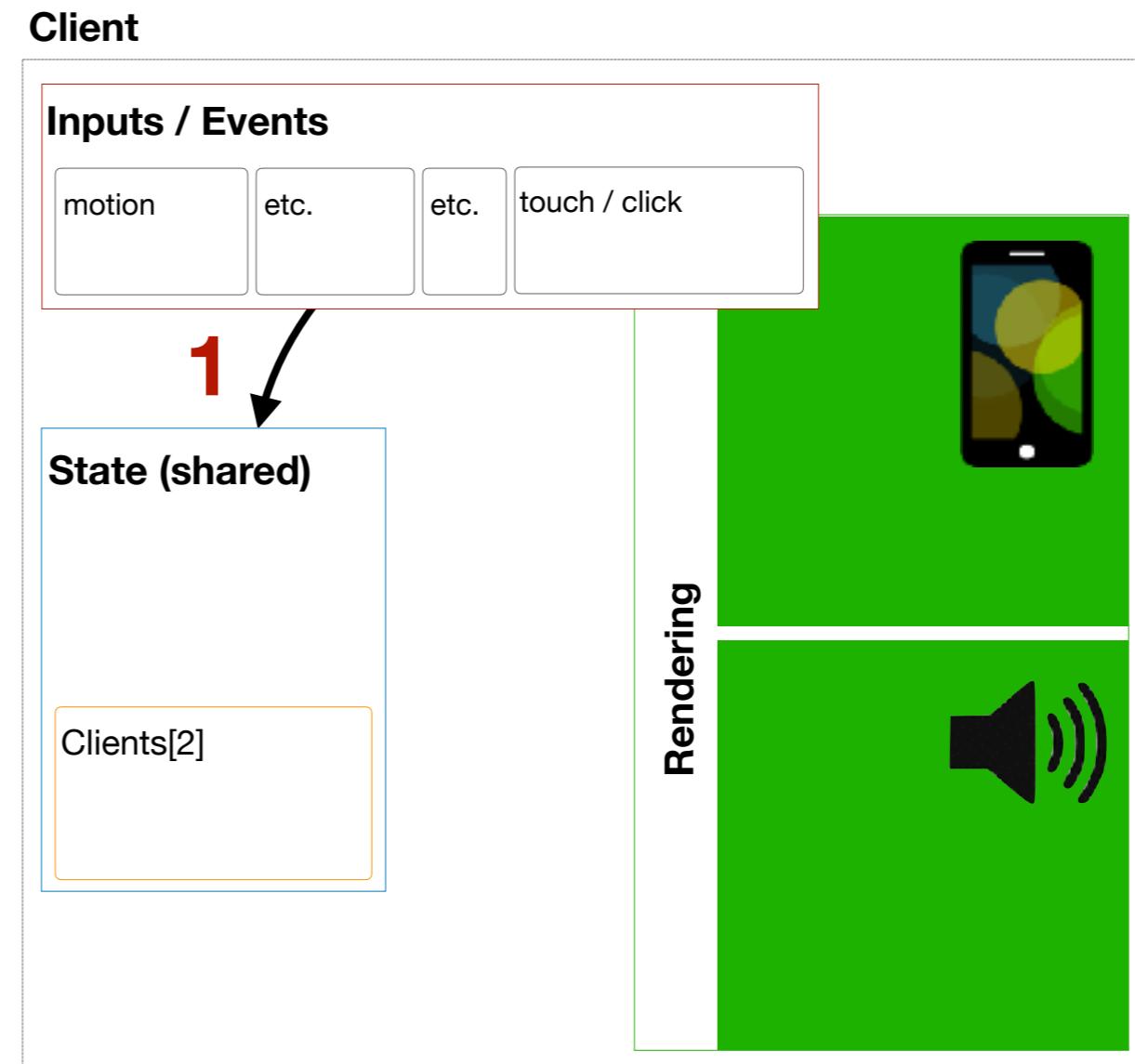
# OVERVIEW



# INITIALIZATION



# STATE MANAGEMENT



# STATE MANAGEMENT

