**Contact:** hongchan@google.com

**Date**: 22 October 2021

**Title**: TBD

**Event**: W3C/SMPTE Joint Workshop on Professional Media Production on the Web

https://www.w3.org/2021/03/media-production-workshop/

**Type**: External/Public

**Notes**:

# Building audio apps on the web

*: thoughts and considerations*

Hongchan Choi

2021.10.22. W3C/SMPTE Joint Workshop on Professional Media Production on the Web

**Hongchan Choi**, PhD

Tech Lead, Google Chrome

W3C Audio WG Co-chair

Lecturer, Stanford University

"What are the ***things*** that

you ***need to think*** about

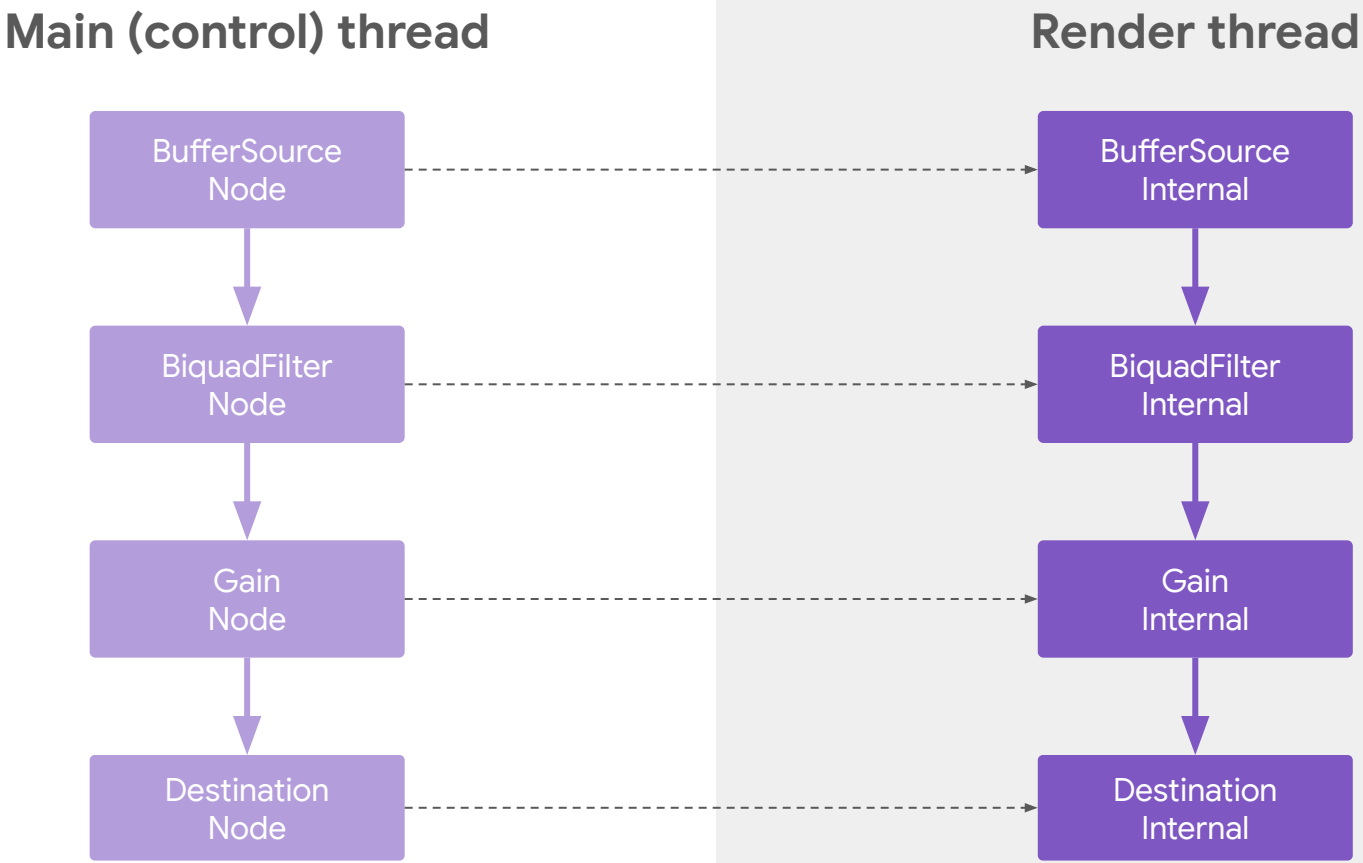
if you were to build

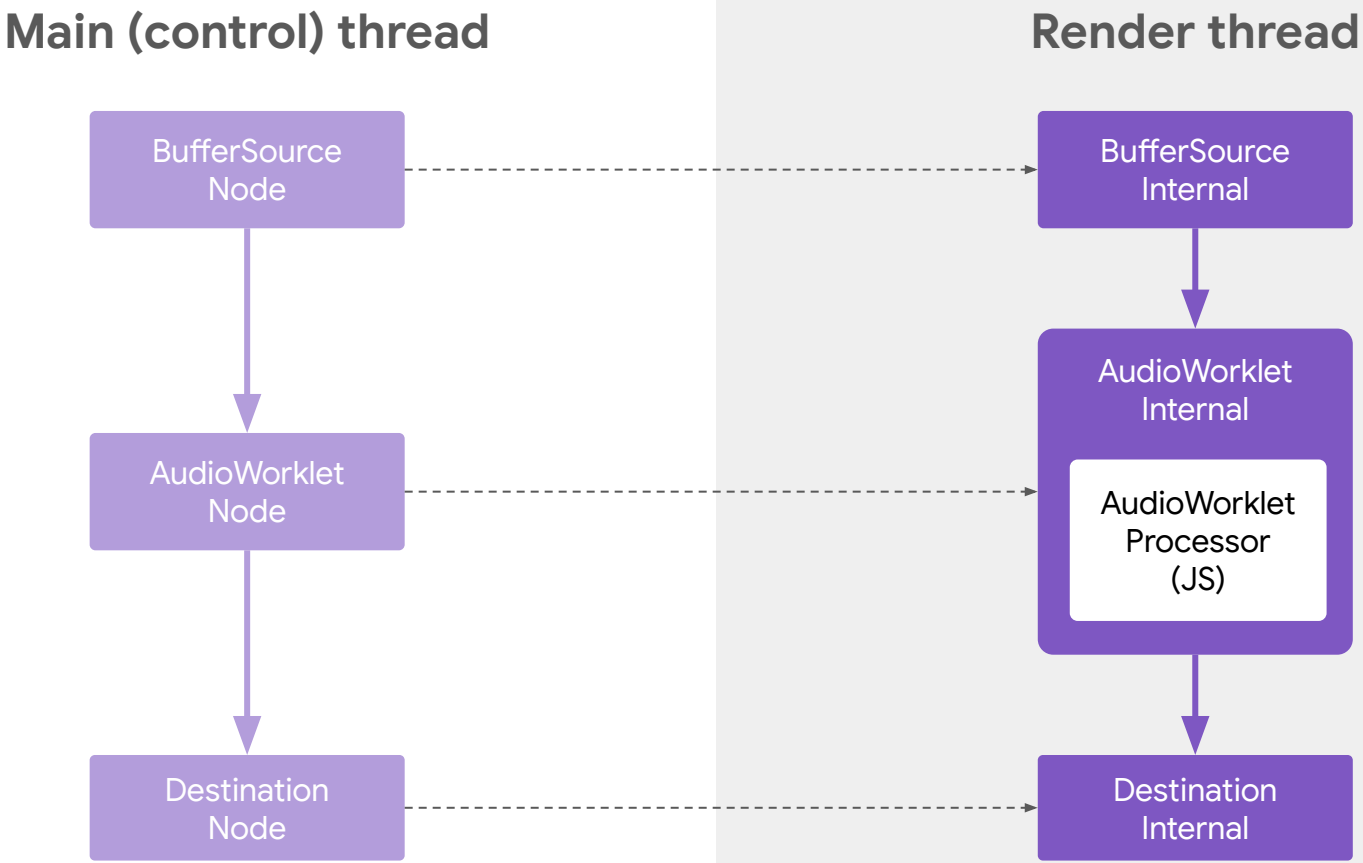
an ***audio app on the web*** today?"

# Web Audio API

# Web Audio API

- A graph-based audio programming environment
- Dual-thread architecture:
    - The renderer runs on a dedicated, high-priority thread.


Main (control) thread
Render thread
BufferSource Node
BiquadFilter Node
Gain Node
Destination Node
BufferSource Internal
BiquadFilter Internal
Gain Internal
Destination Internal

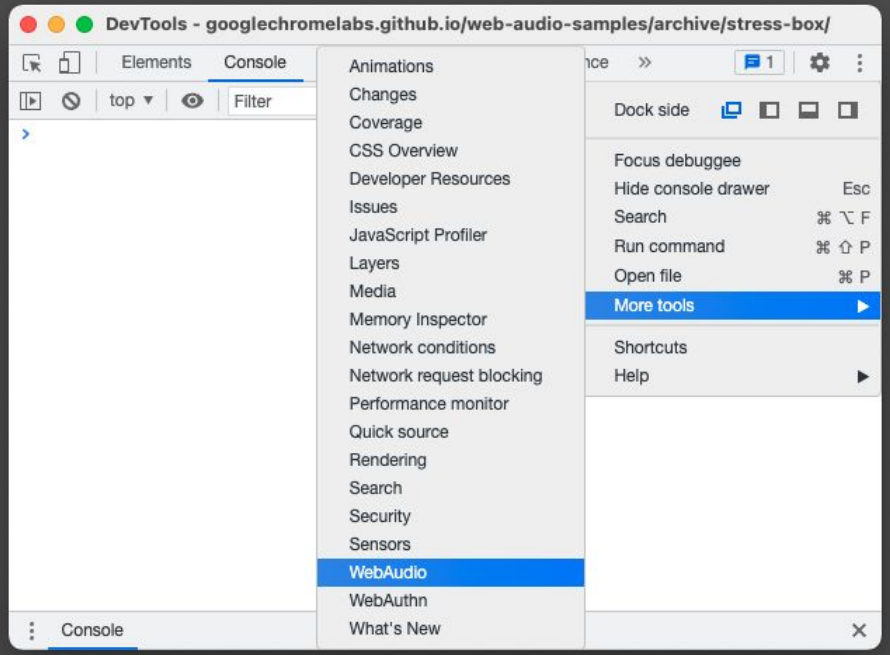

Main (control) thread
Render thread
BufferSource Node
AudioWorklet Node
Destination Node
BufferSource Internal
AudioWorklet Internal
AudioWorklet Processor (JS)
Destination Internal

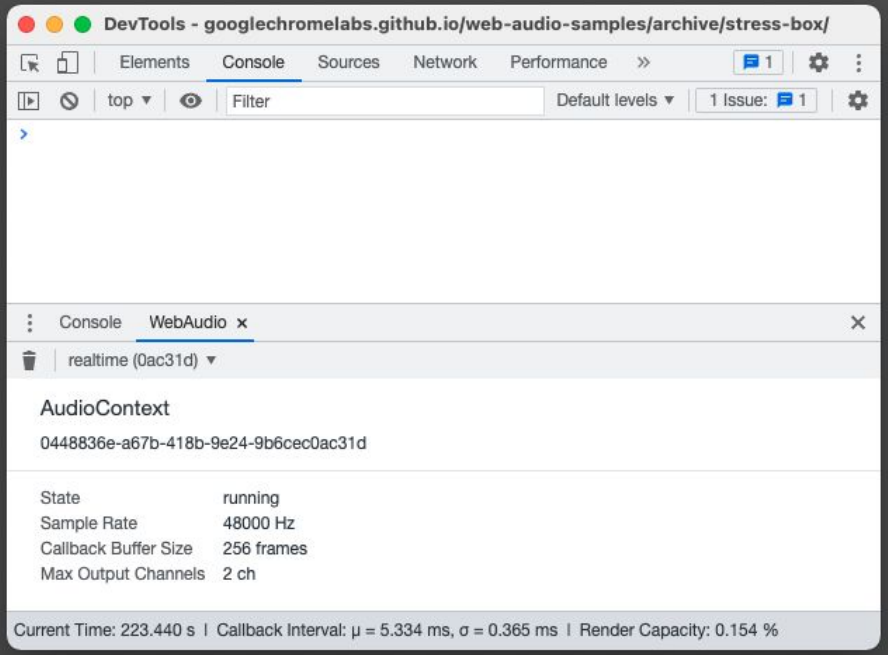
# Web Audio API and GC

- You can't control, but need to care.
- AudioNodes are GC-ed objects.
  - Internals are not, but they still are associated.

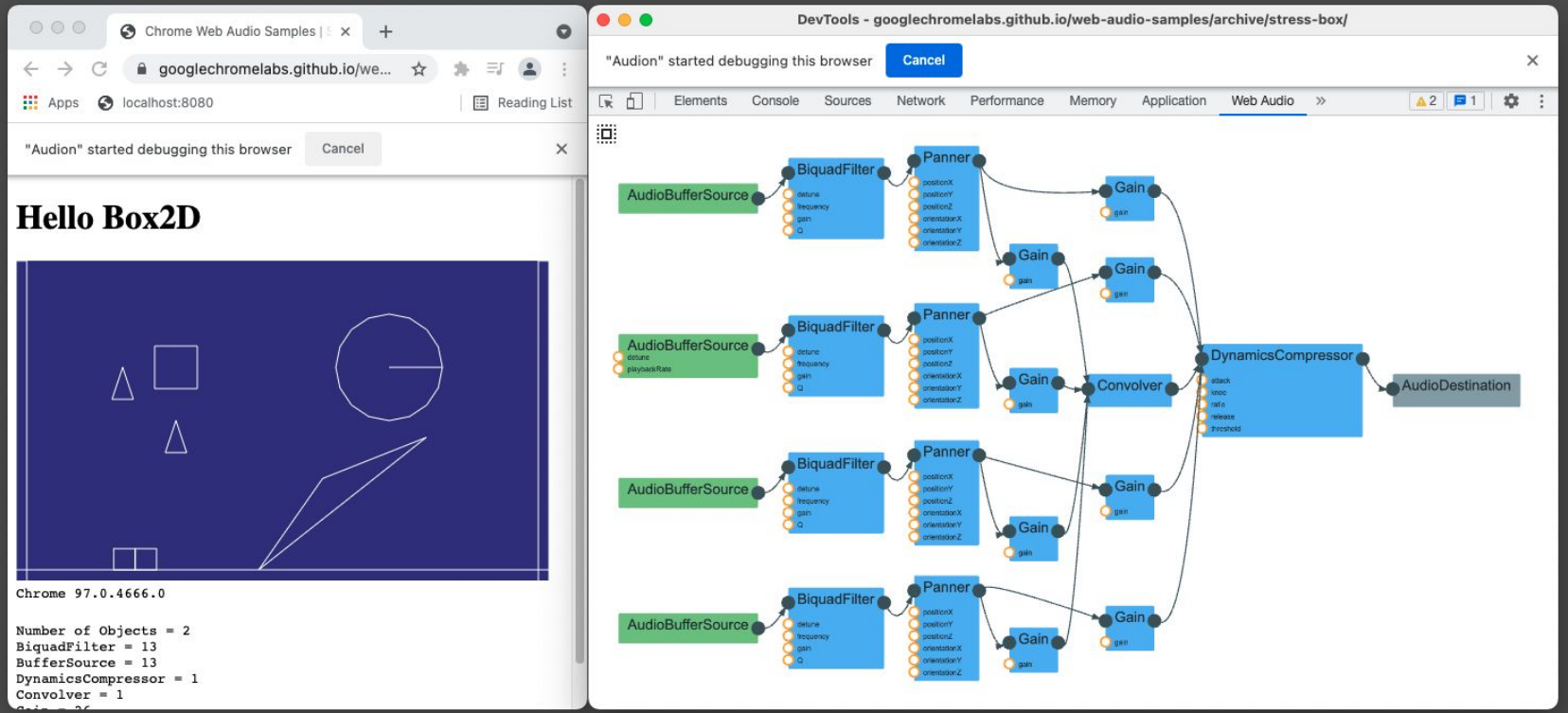
# Web Audio Perf Toolkit

*: Inspect and profile!*

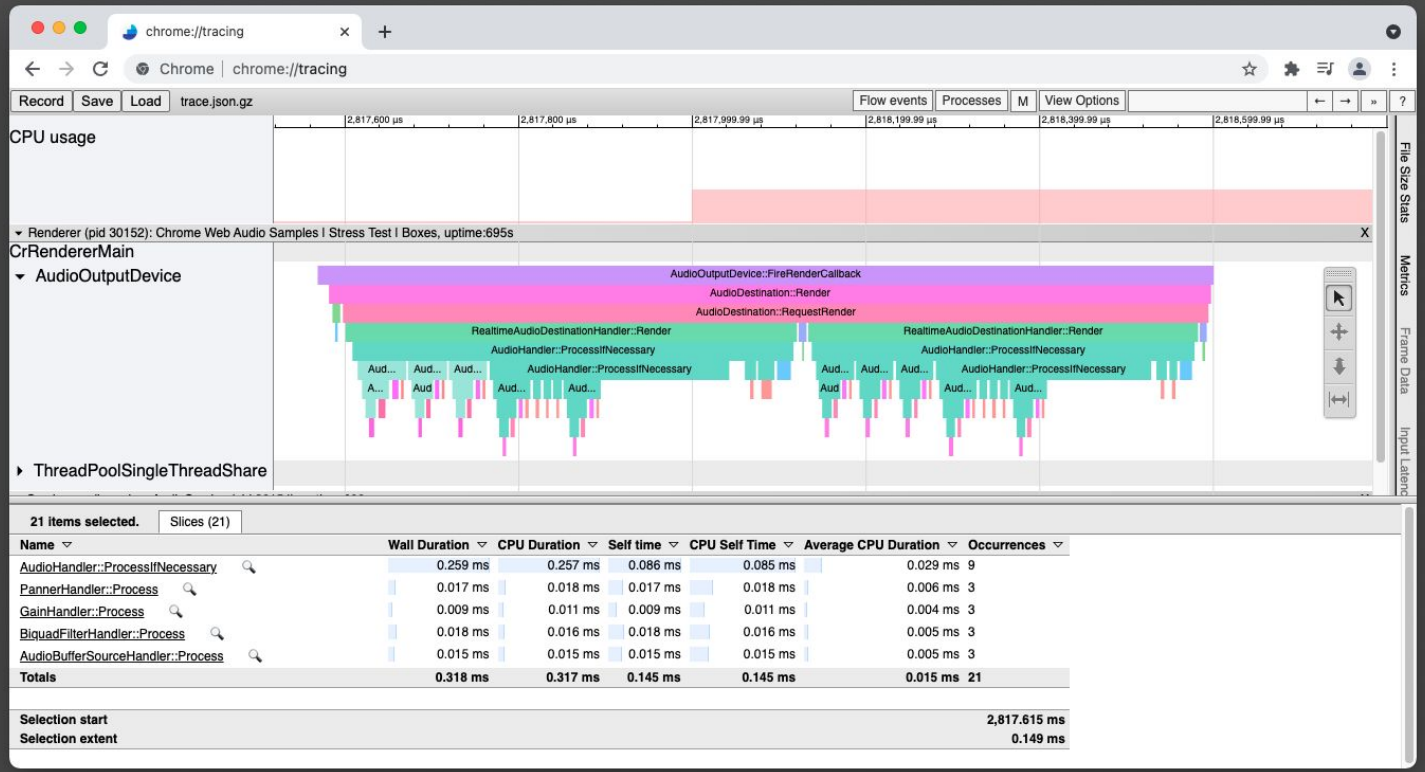



# Web Audio DevTools Panel




# Web Audio Graph Visualizer Extension




# Chrome Tracing

# Other concerns

Device, latency, and user privacy

# 🤔 Device access → User privacy

- Device properties: a rich target of "fingerprinting"
- Mitigation: a constraint-based pattern
  - Protects client information from "drive-by" data collection.
- Native platforms are also gradually changing to adopt similar concepts.

# 🤔 Latency in Web Audio API

- It's important because of
  - Responsive recording and near-realtime monitoring
  - Latency compensation

# 🤔 Latency in Web Audio API

- It's important because of
  - Responsive recording and near-realtime monitoring
  - Latency compensation
- But it's tricky because it's platform/device dependent.
- In Chrome, Web Audio shares infrastructure with RTC/Media.
  - Resilience VS low-latency

# 🤔 Latency: today

- **Input**: getUserMedia()
- **Output**: a system default audio device
- Alternate output devices
    - Need to route Web Audio's output to an AudioElement via MediaStream (i.e. more buffering)

# 🤔 Latency: future

- **Input**:
  - Perhaps AudioContext.selectInputDevice()?
- **Output**:
  - AudioContextOptions.sinkId
  - AudioContext.setSinkId(deviceID)
- Audio WG needs your feedback!

# Conclusion

# Summary

- The design and the architecture of Web Audio API
- Web Audio perf toolkit
- Concerns in device access, latency, and user privacy
- Audio WG needs your feedback!

# Thank you!

bit.ly/webaudio-survey-2021

 @hochsays

 hongchan@google.com