

Csound on the Web

LAC2014

Victor Lazzarini, Edward Costello, Steven Yi and John ffitch

National University of Ireland, Maynooth

LAC Karlsruhe, May 2014

Introduction

- Context of Work
- Goals of Work
- Emscripten
- PNaCl
- Demonstrations
- Conclusions

Csound Ecosystem

- **Command Line - Use Csound as Compiler**
- Desktop Applications - Shell
- Desktop Applications - API
- Mobile Applications - iOS, Android
- Embedded - Raspberry Pi, BeagleBone
- Web - Server-side
- **Web - Client-Side**

Csound Ecosystem

- Command Line - Use Csound as Compiler
- Desktop Applications - Shell
- Desktop Applications - API
- Mobile Applications - iOS, Android
- Embedded - Raspberry Pi, BeagleBone
- Web - Server-side
- **Web - Client-Side**

Csound Ecosystem

- Command Line - Use Csound as Compiler
- Desktop Applications - Shell
- Desktop Applications - API
- Mobile Applications - iOS, Android
- Embedded - Raspberry Pi, BeagleBone
- Web - Server-side
- **Web - Client-Side**

Csound Ecosystem

- Command Line - Use Csound as Compiler
- Desktop Applications - Shell
- Desktop Applications - API
- Mobile Applications - iOS, Android
- Embedded - Raspberry Pi, BeagleBone
- Web - Server-side
- **Web - Client-Side**

Csound Ecosystem

- Command Line - Use Csound as Compiler
- Desktop Applications - Shell
- Desktop Applications - API
- Mobile Applications - iOS, Android
- Embedded - Raspberry Pi, BeagleBone
- Web - Server-side
- **Web - Client-Side**

Csound Ecosystem

- Command Line - Use Csound as Compiler
- Desktop Applications - Shell
- Desktop Applications - API
- Mobile Applications - iOS, Android
- Embedded - Raspberry Pi, BeagleBone
- Web - Server-side
- **Web - Client-Side**

Csound Ecosystem

- Command Line - Use Csound as Compiler
- Desktop Applications - Shell
- Desktop Applications - API
- Mobile Applications - iOS, Android
- Embedded - Raspberry Pi, BeagleBone
- Web - Server-side
- **Web - Client-Side**

Csound on the Web - Goals

- Build client-side web applications using common browser technology
- No extra installation required \Rightarrow Simplified Deployment
- Expand return on investment in knowing Csound
- New way to share work
- Preserve works (dependencies are self-contained)
- Build using main Csound source code, always in sync with latest code changes

Csound on the Web - Goals

- Build client-side web applications using common browser technology
- No extra installation required \Rightarrow Simplified Deployment
- Expand return on investment in knowing Csound
- New way to share work
- Preserve works (dependencies are self-contained)
- Build using main Csound source code, always in sync with latest code changes

Csound on the Web - Goals

- Build client-side web applications using common browser technology
- No extra installation required \Rightarrow Simplified Deployment
- Expand return on investment in knowing Csound
- New way to share work
- Preserve works (dependencies are self-contained)
- Build using main Csound source code, always in sync with latest code changes

Csound on the Web - Goals

- Build client-side web applications using common browser technology
- No extra installation required \Rightarrow Simplified Deployment
- Expand return on investment in knowing Csound
- New way to share work
- Preserve works (dependencies are self-contained)
- Build using main Csound source code, always in sync with latest code changes

Csound on the Web - Goals

- Build client-side web applications using common browser technology
- No extra installation required \Rightarrow Simplified Deployment
- Expand return on investment in knowing Csound
- New way to share work
- Preserve works (dependencies are self-contained)
- Build using main Csound source code, always in sync with latest code changes

Csound on the Web - Goals

- Build client-side web applications using common browser technology
- No extra installation required \Rightarrow Simplified Deployment
- Expand return on investment in knowing Csound
- New way to share work
- Preserve works (dependencies are self-contained)
- Build using main Csound source code, always in sync with latest code changes

Web Technologies

- NPAPI - Going out of use
- Flash - No option for integrating Csound, going out of use
- Silverlight - No option for integrating Csound, going out of use, not cross-platform
- Java Applet/Webstart - Going out of use
- JavaScript/WebAudio - HTML5 Standard, can use Emscripten compiler to compile C/C++ to Javascript
- Portable NativeClient (PNaCl) - works across Operating Systems, requires Chrome/Chromium/Chrome OS with PNaCl support

Web Technologies

- NPAPI - Going out of use
- Flash - No option for integrating Csound, going out of use
- Silverlight - No option for integrating Csound, going out of use, not cross-platform
- Java Applet/Webstart - Going out of use
- JavaScript/WebAudio - HTML5 Standard, can use Emscripten compiler to compile C/C++ to Javascript
- Portable NativeClient (PNaCl) - works across Operating Systems, requires Chrome/Chromium/Chrome OS with PNaCl support

Web Technologies

- NPAPI - Going out of use
- Flash - No option for integrating Csound, going out of use
- Silverlight - No option for integrating Csound, going out of use, not cross-platform
- Java Applet/Webstart - Going out of use
- JavaScript/WebAudio - HTML5 Standard, can use Emscripten compiler to compile C/C++ to Javascript
- Portable NativeClient (PNaCl) - works across Operating Systems, requires Chrome/Chromium/Chrome OS with PNaCl support

Web Technologies

- NPAPI - Going out of use
- Flash - No option for integrating Csound, going out of use
- Silverlight - No option for integrating Csound, going out of use, not cross-platform
- Java Applet/Webstart - Going out of use
- JavaScript/WebAudio - HTML5 Standard, can use Emscripten compiler to compile C/C++ to Javascript
- Portable NativeClient (PNaCl) - works across Operating Systems, requires Chrome/Chromium/Chrome OS with PNaCl support

Web Technologies

- NPAPI - Going out of use
- Flash - No option for integrating Csound, going out of use
- Silverlight - No option for integrating Csound, going out of use, not cross-platform
- Java Applet/Webstart - Going out of use
- JavaScript/WebAudio - HTML5 Standard, can use Emscripten compiler to compile C/C++ to Javascript
- Portable NativeClient (PNaCl) - works across Operating Systems, requires Chrome/Chromium/Chrome OS with PNaCl support

Web Technologies

- NPAPI - Going out of use
- Flash - No option for integrating Csound, going out of use
- Silverlight - No option for integrating Csound, going out of use, not cross-platform
- Java Applet/Webstart - Going out of use
- JavaScript/WebAudio - HTML5 Standard, can use Emscripten compiler to compile C/C++ to Javascript
- Portable NativeClient (PNaCl) - works across Operating Systems, requires Chrome/Chromium/Chrome OS with PNaCl support

Introduction

- Created by Alon Zakai at Mozilla
- Transcompiles LLVM bytecode into Javascript
- Allowing software written in C or C++ to be compiled into Javascript applications for the web
- Uses web standards, no need for plugins

Introduction

- Created by Alon Zakai at Mozilla
- Transcompiles LLVM bytecode into Javascript
- Allowing software written in C or C++ to be compiled into Javascript applications for the web
- Uses web standards, no need for plugins

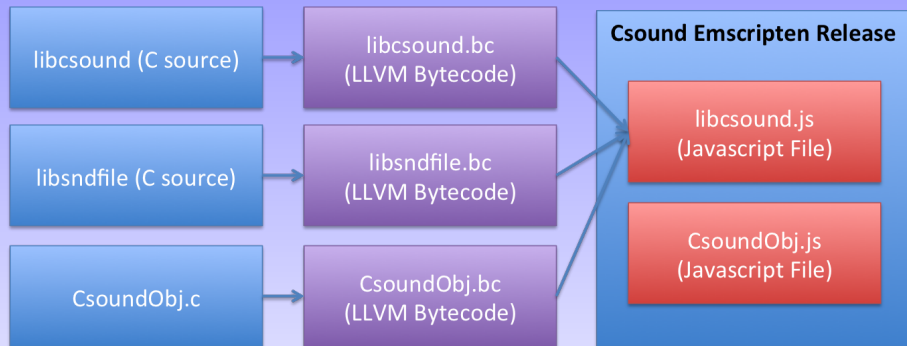
Introduction

- Created by Alon Zakai at Mozilla
- Transcompiles LLVM bytecode into Javascript
- Allowing software written in C or C++ to be compiled into Javascript applications for the web
- Uses web standards, no need for plugins

Introduction

- Created by Alon Zakai at Mozilla
- Transcompiles LLVM bytecode into Javascript
- Allowing software written in C or C++ to be compiled into Javascript applications for the web
- Uses web standards, no need for plugins

Csound Emscripten



Csound Emscripten

- Allows the compilation of .csd, .orc, .sco files
- Csound may be controlled using software input channels, using webpage widgets or OSC; Web MIDI not yet supported in many browsers
- Live audio input is available, as is audio file input
- Potential to access full Csound API

Csound Emscripten

- Allows the compilation of .csd, .orc, .sco files
- Csound may be controlled using software input channels, using webpage widgets or OSC; Web MIDI not yet supported in many browsers
- Live audio input is available, as is audio file input
- Potential to access full Csound API

Csound Emscripten

- Allows the compilation of .csd, .orc, .sco files
- Csound may be controlled using software input channels, using webpage widgets or OSC; Web MIDI not yet supported in many browsers
- Live audio input is available, as is audio file input
- Potential to access full Csound API

Csound Emscripten

- Allows the compilation of .csd, .orc, .sco files
- Csound may be controlled using software input channels, using webpage widgets or OSC; Web MIDI not yet supported in many browsers
- Live audio input is available, as is audio file input
- Potential to access full Csound API

Introduction

- Open-source, Developed by Google
- Architecture Independent form of Native Client, which allows running native code in a sandbox
- Compiles C/C++ into abstract, architecture-independent peexe executables
- Bi-directional communication with JavaScript via the Pepper API
- PNaCl is available in Google's Chrome, Chromium, and Chrome OS

Introduction

- Open-source, Developed by Google
- Architecture Independent form of Native Client, which allows running native code in a sandbox
- Compiles C/C++ into abstract, architecture-independent peexe executables
- Bi-directional communication with JavaScript via the Pepper API
- PNaCl is available in Google's Chrome, Chromium, and Chrome OS

Introduction

- Open-source, Developed by Google
- Architecture Independent form of Native Client, which allows running native code in a sandbox
- Compiles C/C++ into abstract, architecture-independent peexe executables
- Bi-directional communication with JavaScript via the Pepper API
- PNaCl is available in Google's Chrome, Chromium, and Chrome OS

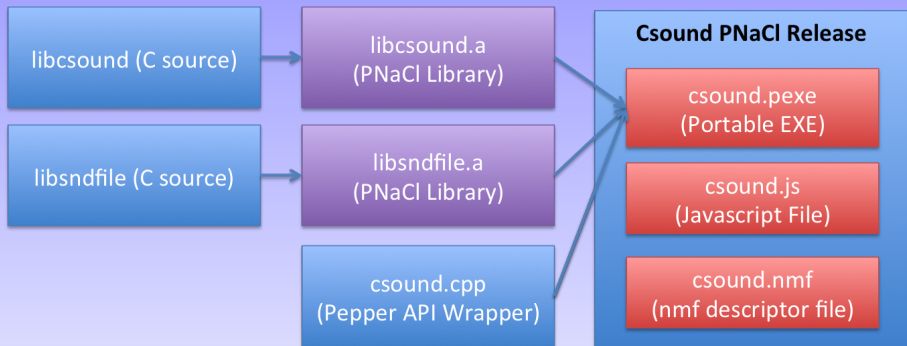
Introduction

- Open-source, Developed by Google
- Architecture Independent form of Native Client, which allows running native code in a sandbox
- Compiles C/C++ into abstract, architecture-independent peexe executables
- Bi-directional communication with JavaScript via the Pepper API
- PNaCl is available in Google's Chrome, Chromium, and Chrome OS

Introduction

- Open-source, Developed by Google
- Architecture Independent form of Native Client, which allows running native code in a sandbox
- Compiles C/C++ into abstract, architecture-independent peexe executables
- Bi-directional communication with JavaScript via the Pepper API
- PNaCl is available in Google's Chrome, Chromium, and Chrome OS

Csound PNaCl



Csound PNaCl API

- Play and Pause
- Compile ORC and SCO code
- Send score messages
- Send values over channels
- Not full access to Csound API, similar to `csound~` and `csoundapi~`
- No audio input (yet, Pepper API Beta has added support)

Csound PNaCl API

- Play and Pause
- Compile ORC and SCO code
- Send score messages
- Send values over channels
- Not full access to Csound API, similar to `csound~` and `csoundapi~`
- No audio input (yet, Pepper API Beta has added support)

Csound PNaCl API

- Play and Pause
- Compile ORC and SCO code
- Send score messages
- Send values over channels
- Not full access to Csound API, similar to `csound~` and `csoundapi~`
- No audio input (yet, Pepper API Beta has added support)

Csound PNaCl API

- Play and Pause
- Compile ORC and SCO code
- Send score messages
- Send values over channels
- Not full access to Csound API, similar to `csound~` and `csoundapi~`
- No audio input (yet, Pepper API Beta has added support)

Csound PNaCl API

- Play and Pause
- Compile ORC and SCO code
- Send score messages
- Send values over channels
- Not full access to Csound API, similar to `csound~` and `csoundapi~`
- No audio input (yet, Pepper API Beta has added support)

Csound PNaCl API

- Play and Pause
- Compile ORC and SCO code
- Send score messages
- Send values over channels
- Not full access to Csound API, similar to `csound~` and `csoundapi~`
- No audio input (yet, Pepper API Beta has added support)

Demonstrations

- Emscripten and PNaCl Test Suites
- Web Sequencer (PNaCl)
- Csound Notebook (PNaCl)
- Processing.js Example (PNaCl)
- Manual Example (Emscripten)
- OSC Demo (Emscripten)

Emscripten and PNaCl Test Suites

Emscripten: <http://eddyg.github.io/CsoundEmscripten>

PNaCl: <http://vlazzarini.github.io>

Web Sequencer

<http://fcahoon.github.io/seq/>

Csound Notebook

<http://csound-notebook.kunstmusik.com>

ProcessingJS Example

http://www.kunstmusik.com/processingjs_example

Manual Example

<http://eddyc.github.io/CsoundManual>

OSC Demo

<http://eddyc.github.io/CsoundEmscripten>

Conclusions

- Emscripten: More ubiquitous, less performant, faster loading time, limited by Web Audio API
- PNaCl: More performant, less ubiquitous, design limitations with message passing
- Both technologies are actively developed

Future Work

- Continue to develop the two web API's based on user requirements
- Look at unified API that can use either PNaCl or Emscripten builds, depending on what is available (also keep an eye on PepperJS)

Acknowledgements

This work has been partly funded by the Irish HEA PRTL-5 Digital Arts and Humanities programme.