Collabora Online: WASM

Caolán McNamara

caolan.mcnamara@collabora.com

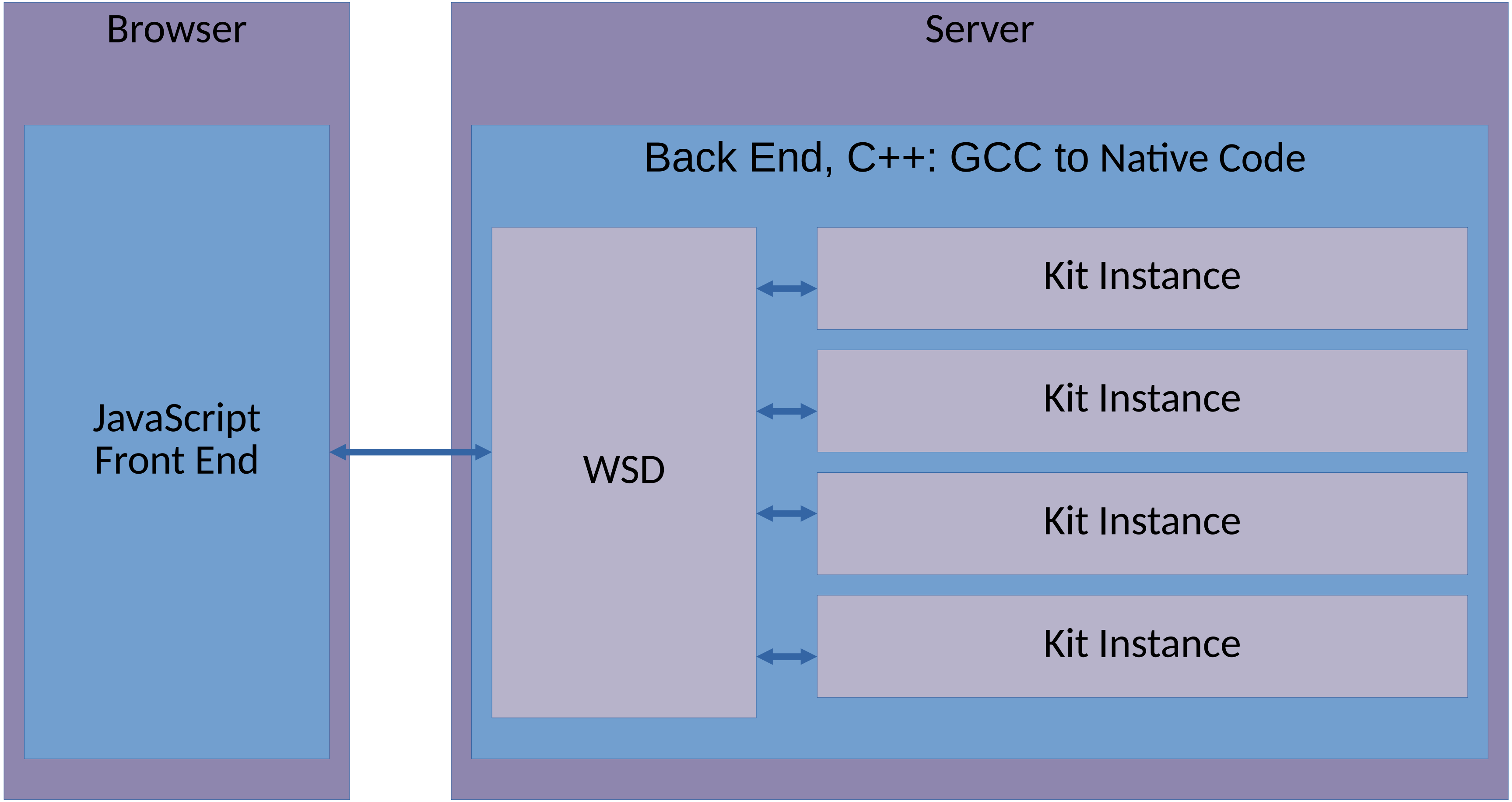
Thorsten Behrens

thorsten.behrens@allotropia.de

Collabora Online

FOSDEM

Collabora Online: Typical Overview



Browser

JavaScript Front End

Server

Back End, C++: GCC to Native Code

WSD

Kit Instance

Kit Instance

Kit Instance

Kit Instance

Collabora Online

No server and it doesn’t work of course

Kit Instance

The big binary piece

- Links to LibreOffice core
- du -ch of all core shared libs is 317MB
- One instance per document
- Server mediates between browser javascript clients and kit, forwarding client requests to kit and tiles, etc from kit to clients

Portability

Core Ports

- OS: Linux, Windows, macOS, iOS, Android, *BSD, etc
- UNO ABI Archs: x86, x86_64, aarch64, alpha, hppa, ia64, m68k, mips[64], power[64], s390[x], sparc[64], etc

Collabora Online Ports

- Linux, iOS, Android, *BSD, etc
- Less low level ABI requirements

Web Assembly

High performance binary executable format

- Available in browsers for years
  - Runs in the same sandbox as JavaScript
- Emscripten compiles C++ to WASM with LLVM
- Website Security Policy determines if it is allowed to be executed

LibreOffice WASM Port

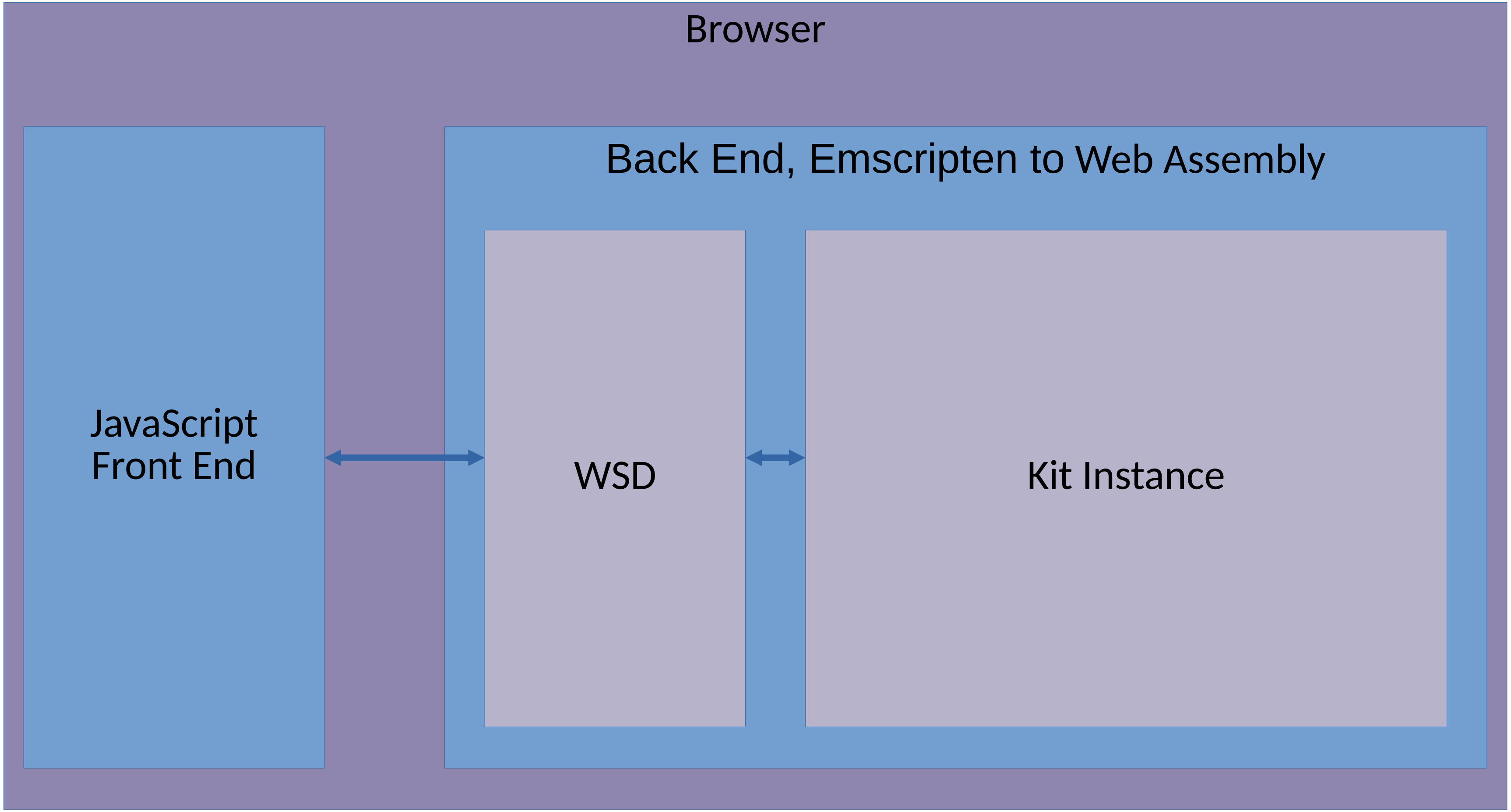
allotropia WASM Port

- Port of LibreOffice to WebAssembly aka WASM using the Emscripten toolchain.

    - https://wiki.documentfoundation.org/Development/WASM
- Gory details on porting to WASM
    - https://git.libreoffice.org/core/+/refs/heads/master/static/README.wasm.md

Collabora Online

Collabora Online: COWASM Overview



Browser

JavaScript Front End

Back End, Emscripten to Web Assembly

WSD

Kit Instance

Collabora Online

COWASM

allotropia WASM Port

- core+online ported to wasm
- Normal online server when requested by client redirects to a wasm page which triggers downloading the wasm binary
  - Get this just right and the wasm can be cached so it’s a one time download
  - Feed it a copy of the document
- COWASM embedded online server executes in the browser
- JavaScript client communicates with embedded COWASM similar to normal server

Collabora Online, Offline: COWASM

Security Policy Headache #1

Collabora Online integration

- Intricate dance of multiple web applications and servers
- SPECTRE
  - https://en.wikipedia.org/wiki/Spectre_(security_vulnerability)
  - So Browsers super paranoid about allowing wasm to execute
  - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/SharedArrayBuffer
- Basically both the embedding app and embedded app have to agree to that arrangement

Security Policy Headache #2

Nextcloud

- Thanks to Julius Härtl for bootstrapping how to get the Nextcloud richdocuments integration to provide the appropriate security content headers from that side
  - https://github.com/nextcloud/richdocuments/pull/3260
- Then can set matching ones from the Collabora Online side to get the browsers to allow WASM
- Not plain sailing yet. Configure CO for reverse-proxy mode, so all data appears from the same server hosting Nextcloud
- And chrome needs https
- And maybe some sites pull logos, etc from a third location that doesn’t have the magic headers
  - So make the chain of adding headers conditional on wasm enabled in Collabora Online

Practicalities

- Build time resources
  - Linking takes > 25G RAM
- Cross compiling
  - Cross compiling is always a little fraught
  - podman pull public.ecr.aws/allotropia/libo-builders/wasm
- Threading
  - A little unclear if number of threads reported in wasm is threads the system has or the number of threads wasm can use
- Incomplete
  - Just one way online→offline for now

But it works

FOSDEM

Collabora Online

Thank you!

T S P D

@CollaboraOffice
hello@collaboraoffice.com
www.collaboraoffice.com