A SNEAK PEEK AT

WASM

By Leo Melin / Gofore Oy

HISTORY

- Mozilla did ASM.js (subset of JS)
- Google did PNaCl (Portable Native Client)
- –> Apple, Mozilla, Microsoft & Google started WebAssembly

WHAT IS IT?

- Assembly-like language for Web
- Specification of bytecode for web -> faster to parse, faster to execute than JS
- ASM.js made right
  - Collaboration with major browser manufacturers
  - Faster decoding
  - Better performance

FEATURES

- Full threading support
- Integrated CG
- DOM integration
- Multi-process support
- Source maps
- + lots of compiler level stuff (SIMD types & intrinsics, coroutines, dynamic linking, tail-call optimization)

IT'S NOT THE END OF JS

WHAT'S IN THE FUTURE?

- HTML/CSS/JS/WASM combinations
- Full HTML/CSS/WASM app
- WASM based frameworks/libraries
- Some extreme experimental sh*t

USE CASES

- Image / video editing
- Games, game portals, heavy 3d
- Peer-to-peer applications (games, collaborative editing, decentralized and centralized)
- Music applications (streaming, caching)
- Video/audio codecs in Wasm
- Image recognition
- Live video manipulation
- VR and augmented reality (very low latency)
- CAD applications
- Visualisation / simulation / emulation / virtual machine
- POSIX user-space environment, allowing porting of existing POSIX applications
- Developer tooling (editors, compilers, debuggers, …)
- Remote desktop
- Encryption/Decryption

USING WASM TODAY

- ASM.js FTW
- You CAN use clang to produce wasm
- Format is still changing
- Not supported very well
- Binaryen toolkit for WASM

WORKFLOW

- 1. Write source in C/C++
- 2. Compile source to .s -format
- 3. .s format –> Wasm text format (s-expression)
- 4. Wasm text format –> Wasm binary format
- 5. const Foo = Wasm.instantiateModule(binary).exports.Foo
  - ...or propably at some point:
    import Foo from ‘foo.wasm’

DEMO

TRY IT OUT!

- Install Binaryen
- Install emscripten
- Install clang
- Do some hacking & experiment!

HAPPY HACKING!

. . . READ MORE:

WebAssembly site

WASM design

Binaryen toolkit

Emscripten

WebAssembly High Level Goals

WebAssembly explained