WebAssembly
as a safe universal platform

Jiří Zárevúcky
zarevucky.jiri@gmail.com (personal)
jiri.zarevucky@nic.cz (work)
Original motivation

- HelenOS: many platforms, including IA64 (Itanium)
- LLVM: no official support for IA64, former support buggy, unmaintained
- Rust: LLVM backend only
- HelenOS with servers implemented in Rust ???

Answer? Build to WebAssembly, translate wasm to pure C.
WebAssembly

- Binary distribution format.
- Virtual ISA.
- Fast, secure client-side code execution.
- Made for web browsers, but much more versatile!
- Stack-based VM, but…
  - Drawbacks usually associated with such VMs don’t apply.
- Strict well-formedness.
- Strict semantic requirements, deterministic (mostly).
- Checked memory access.
- Simple.
- Function import from environment.
Rust

- Safety and performance.
- Difficult to learn, but high rewards.
- LLVM backend, supports WebAssembly
- C ABI interoperability…
  - Translates to imports in wasm!
WebAssembly to C

- Stack slots = local variables, no stack manipulation.
- Most instructions: calls to C shim
  
i64.add → stack_5 = _i64_add(stack_1, stack_4);

- Imported functions → extern C functions
  - (BUT can’t directly pass pointers. Safety!)
- Very simple translator. (< 1000 loc)
  - Crude output, but GCC can optimize it.
- Flexible!