Preventing Over-fragmentation: Specifying Wasm Sublanguages

Andreas Rossberg
Dfinity

Eco system diversity

Wasm is now used in many different eco systems

- Web, CDNs, block chains, enclaves, IoT, …

Fragmentation lurks

Different eco systems have different constraints

Some can't or don't want to use certain features
- threads, GC, SIMD, non-determinism …

This *will* happen!

But if multiple parties subset Wasm,
let's make sure they do it consistently!

Minimising Fragmentation

Idea: Language compartmentalisation

Precisely specify certain sublanguages

- Wasm minus feature set X or Y

Ideally, a very small number of Xs and Ys – tbd

Suggested Approach

Purely syntactic

Annotate AST productions that belong to X

Define Wasm\X = Wasm minus all productions marked X

Implies binary and text format constructs

Spec Example

memtype ::= *limits share*

share ::= unshared

| shared (T)

instr ::= …

| struct.new (G)

| struct.get_*sx*? (G)

| struct.set (G)

| …

(T = threads G = GC)

Process

Not really doable as a proposal in its own right

Has to be done as part of the proposals that are affected

Obvious candidates: threads, SIMD, GC

Thoughts? Poll?

Should the Wasm spec specify language subsets?