



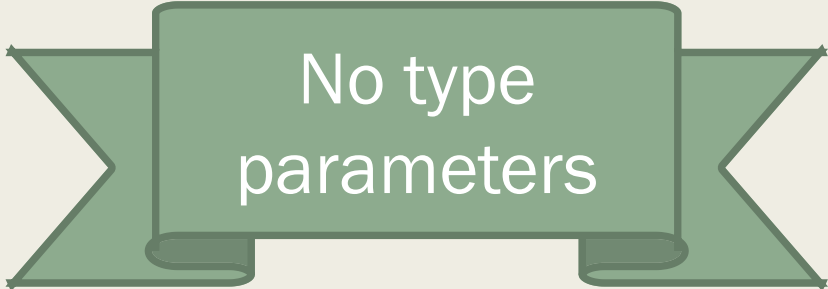
# Lightly Typed WebAssembly

Ross Tate  
(and the SOIL Initiative)



# WebAssembly Value Types

- Numeric
  - *i32, f32, i64, f64, i128*
- Executable
  - *funcref*
- Heap
  - *structref, tagref*
- Stack
  - *stackref*
- Abstract
  - *externref, imported value types*

A green ribbon banner with a dark green border and a slight 3D effect, containing the text "No type parameters".

No type  
parameters

A green ribbon banner with a dark green border and a slight 3D effect, containing the text "No subtyping".

No subtyping

A green ribbon banner with a dark green border and a slight 3D effect, containing the text "No μ/v types".

No  $\mu/v$  types

# Executable Values

formerly  
“dispatch tags”

## funcref with Call Tags

- Slightly less efficient in worst case
  - *push tag, pop tag, compare*
- Optimized common case
  - *no instance-switch for non-exported call tags*
- Flexible dynamism
  - *switching on call tag enables dynamic stack features*

## Typed Function References

Obvious small costs  
(often optimizable)

Non-obvious  
significant savings

Increased flexibility

complex type system

regular-coinductive subtyping

possibly co- and contra-  
type constructors

# Lightly Typed WebAssembly

- Numeric
  - *i32, f32, i64, f64, i128*
- Executable
  - *funcref (call-tags)*
- Heap
  - *structref, tagref (gc#109 w/ JP)*
- Stack
  - *stackref (design#1360 w/ Francis)*
- Abstract
  - *externref, imported value types*
- Easy to implement
  - *gets it out the door quickly*
- Easy to target
  - *facilitates a diverse community*
- Easy to analyze
  - *collect data on if/where/which types would help*
- Easy to extend
  - *can add types, based on data, as needs arise*