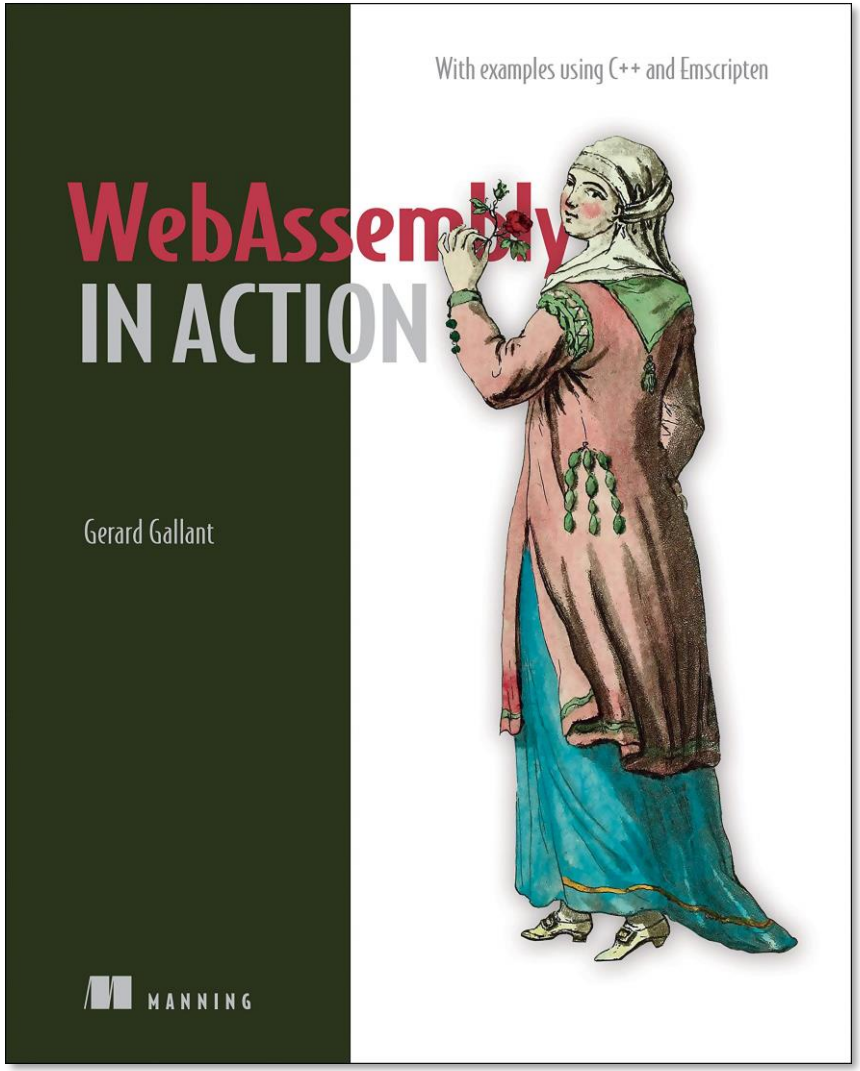
WebAssembly Overview

About Me

- Senior Developer and Architect with Dovico Software
- Runner
- Author of WebAssembly in Action

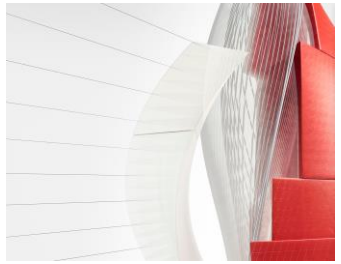


@Gerard_Gallant

Some real-world examples

PSPDFKit

ebay



AutoCAD Web App

CLOUDFLARE Workers

unity

Adobe – View SDK

UNREAL ENGINE

https://madewithwebassembly.com/

Browsers

Performance

JavaScript

Native Client (NaCl)

- C code is compiled ahead of time
- Secure
- Fast

asm.js

- C code is transpiled ahead of time
- Fast code execution
- Fast from the first call

asm.js

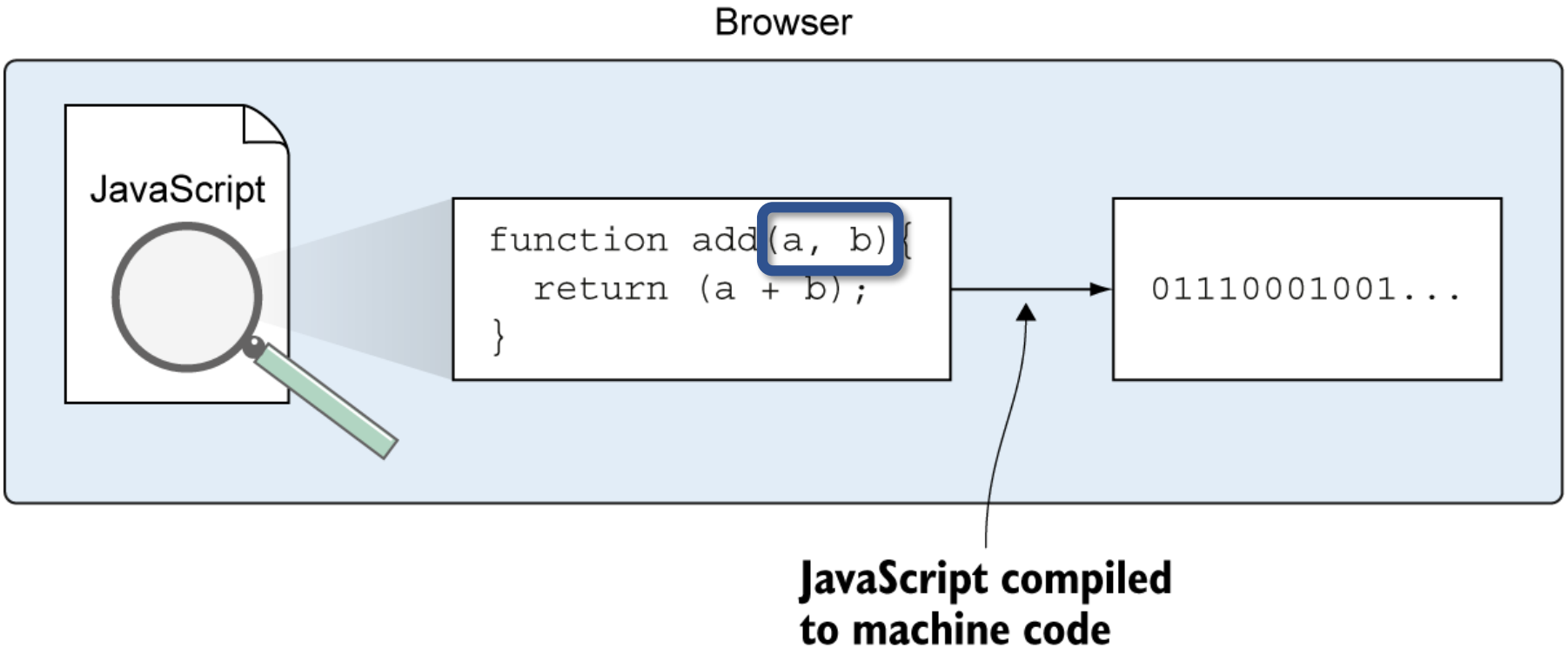
```
function AsmModule() {
    "use asm";
    return {
        add: function(a, b) {
            a = a | 0;
            b = b | 0;
            return (a + b) | 0;
        }
    }
}
```

WebAssembly MVP

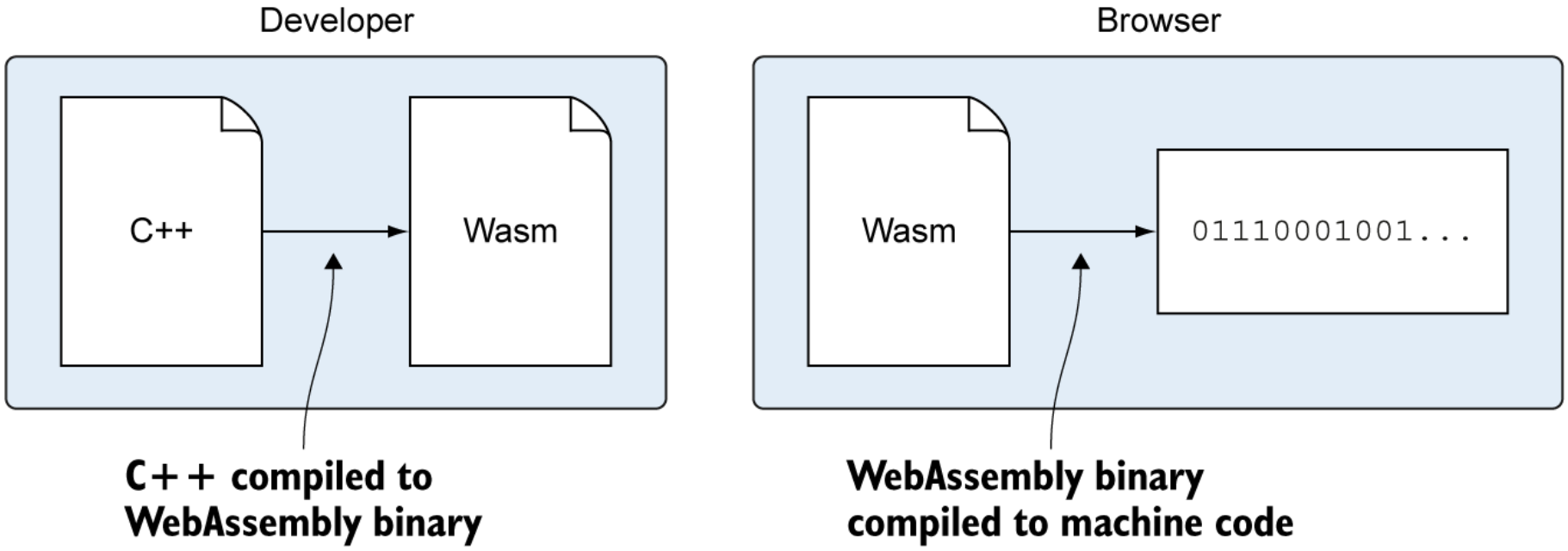
In 2017, all 4 major browsers added support for WebAssembly (Wasm)

- Compiler target
- Fast
- Secure
- Separate from JavaScript
- Portable

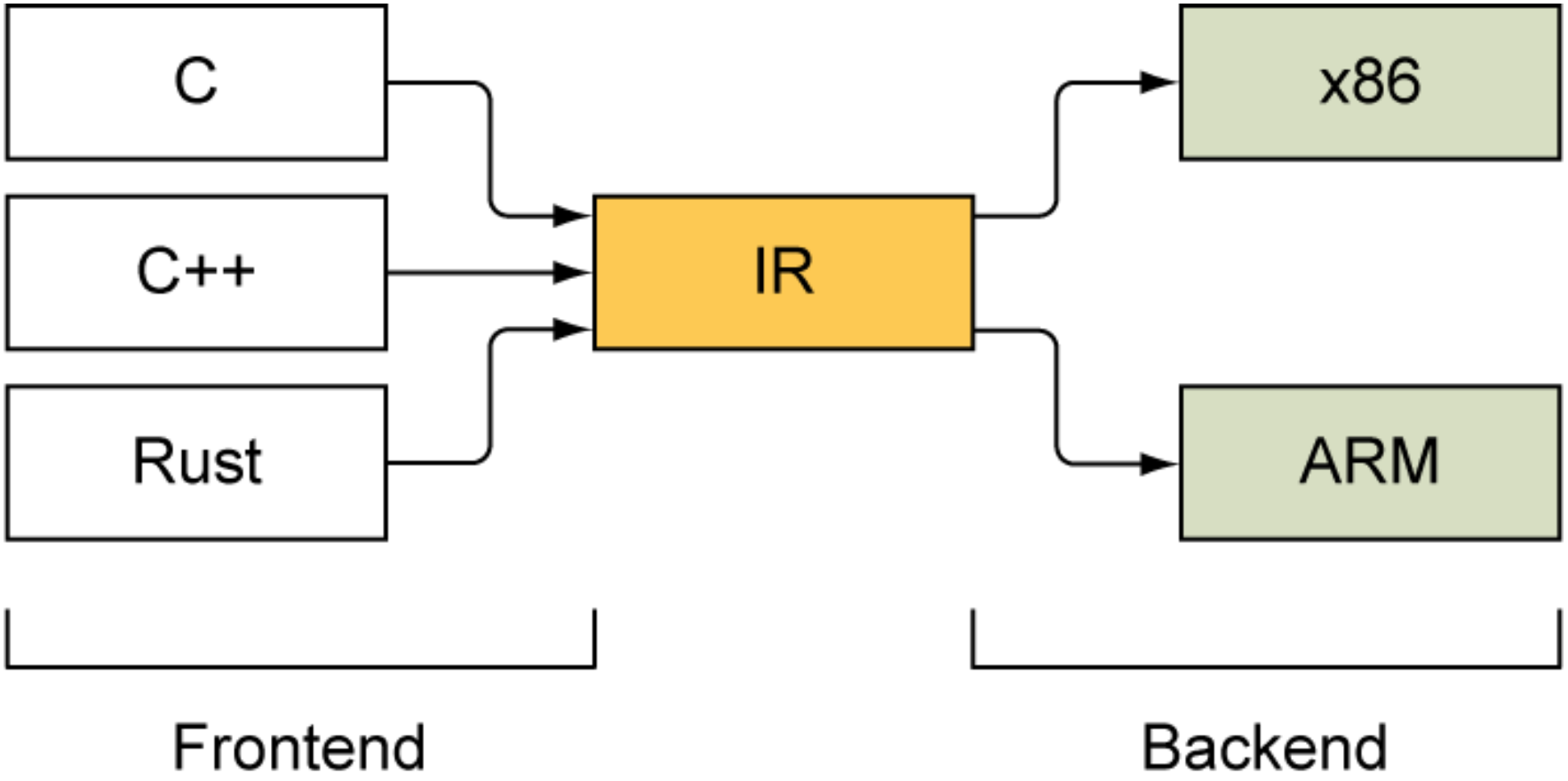
JavaScript


Browser
JavaScript
function add(a, b){
return (a + b);
}
01110001001...
JavaScript compiled
to machine code

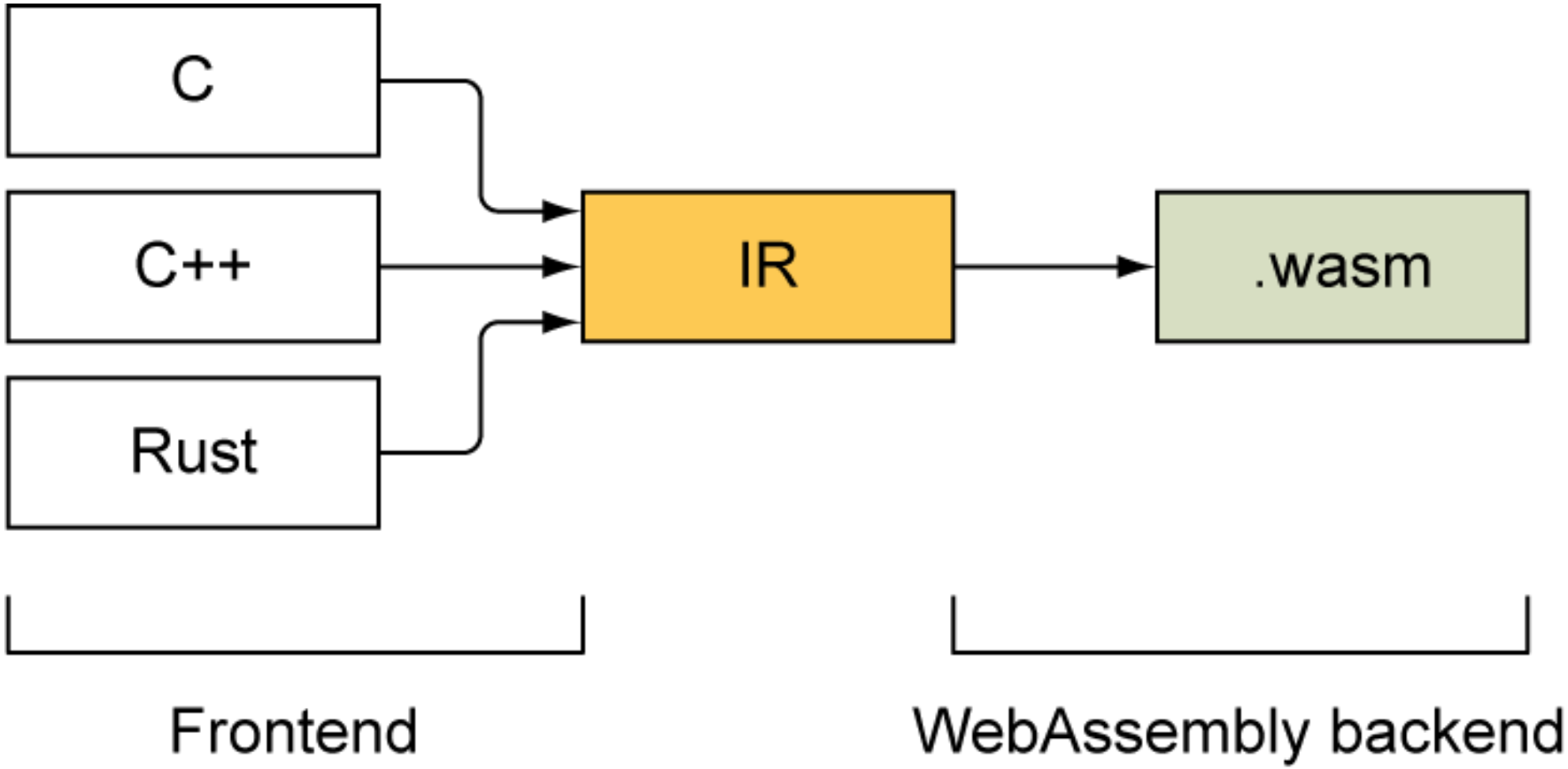
WebAssembly



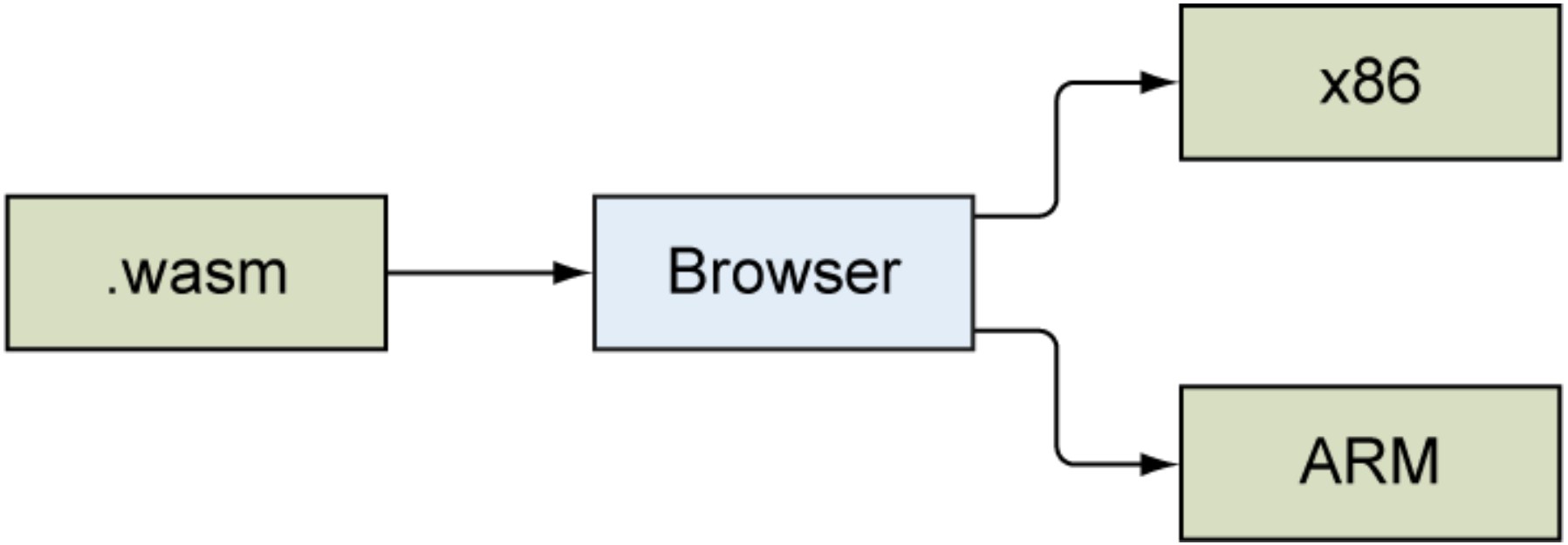
Traditional Compiler



WebAssembly Compiler



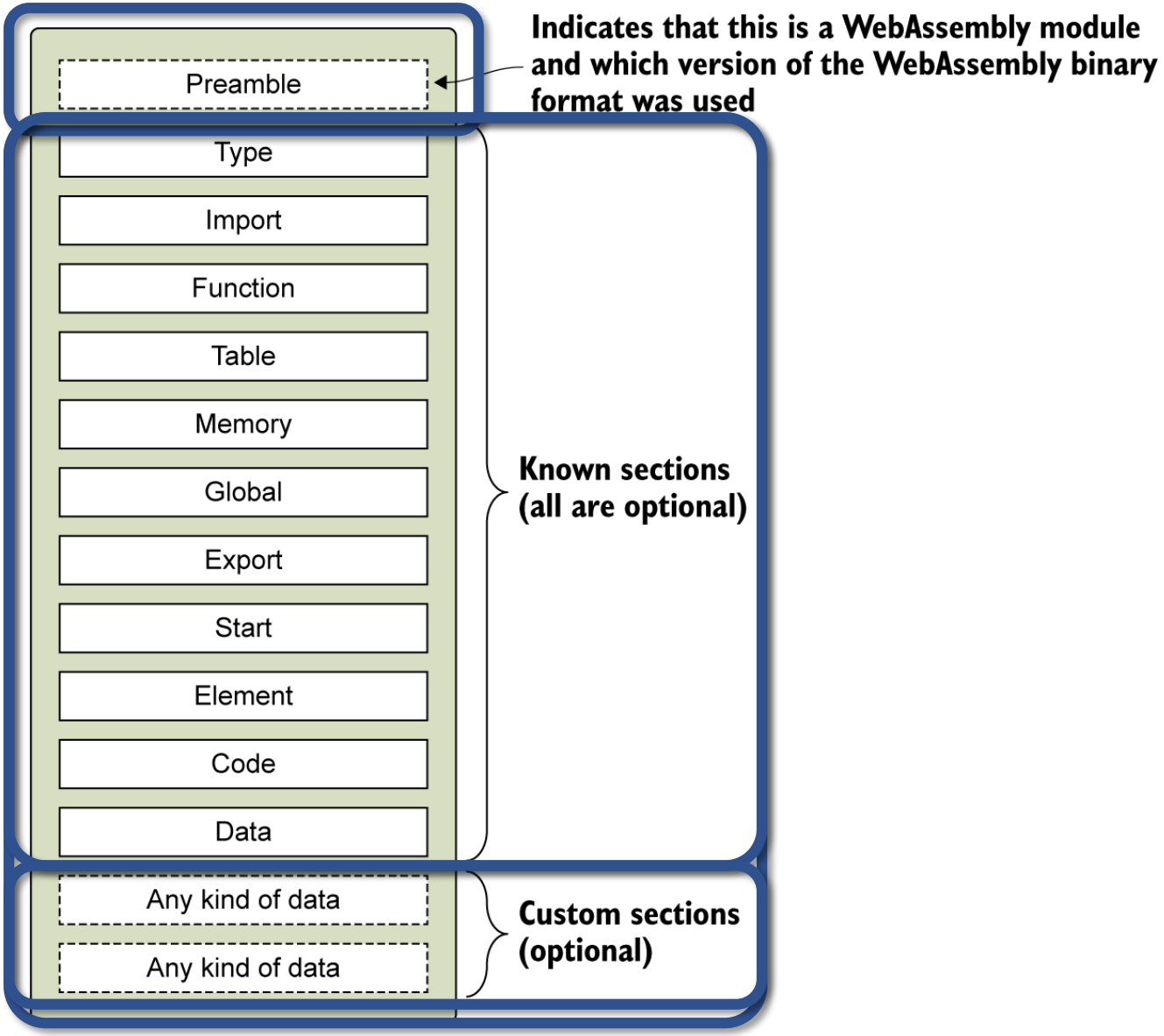
In the Browser



Structure

Value Types:

- 32-bit integer
- 32-bit float
- 64-bit integer
- 64-bit float



What makes it secure?

- The code is validated before it’s compiled
- Runs within the JavaScript VM (not a new VM)
- Same security policies as JavaScript
- No direct access to the device memory
- No direct access to objects like function pointers
- Execution stack is separate and inaccessible to the code

Which languages you can use?

- C and C++
- Rust
- AssemblyScript – TypeScript
- TeaVM – Java
- Go 1.11
- Pyodide – port of the core packages of Python's scientific stack
- Blazor – C#
- The Uno Platform – C#/XAML

https://github.com/appcypher/awesome-wasm-langs

Outside the Browser

WebAssembly System Interface (WASI)

https://hacks.mozilla.org/2019/03/standardizing-wasi-a-webassembly-system-interface/

By Lin Clark

Byte Code Alliance

https://hacks.mozilla.org/2019/11/announcing-the-bytecode-alliance/

By Lin Clark

Outside the Browser

- Node.js
- Edge and Serverless computing
- IoT
- Desktop
- ewasm
- In your code

The Future

WebAssembly proposals:
https://github.com/WebAssembly/proposals

- Direct DOM and Web API access
- Garbage collection
- Exception handling
- Threads
- Better tooling support

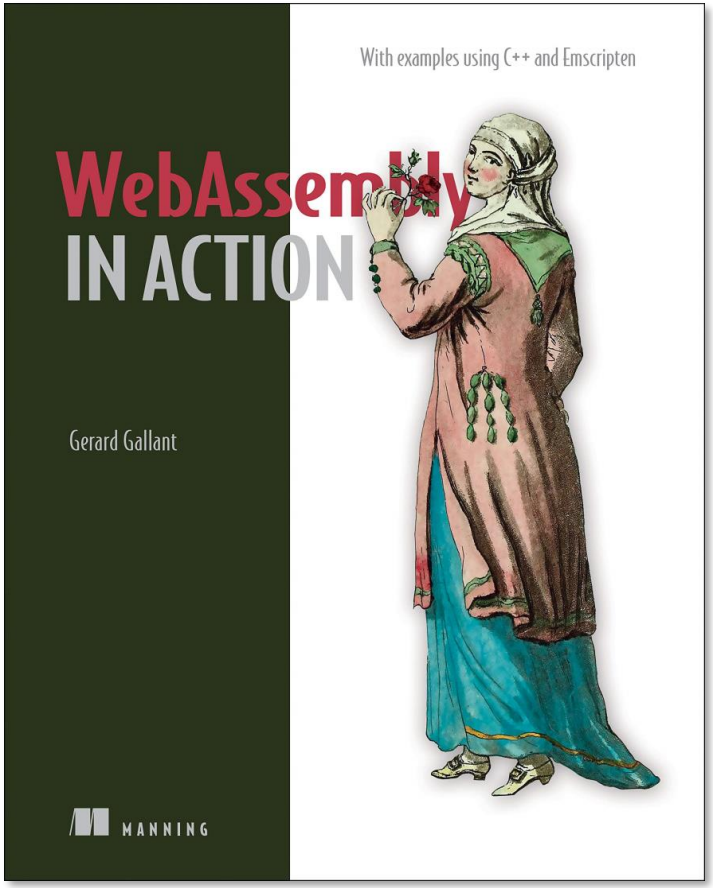
Thank You

40% off all products in all formats:

ctwconfoo20

www.manning.com

@Gerard_Gallant



https://github.com/confooca/yul2020-slides