WebAssembly: The Big Picture

WHAT IS WEBASSEMBLY?

Barry Luijbregts
SOFTWARE ARCHITECT & DEVELOPER

@AzureBarry www.azurebarry.com

Introduction

Where WebAssembly came from

What it is and what it can be used for

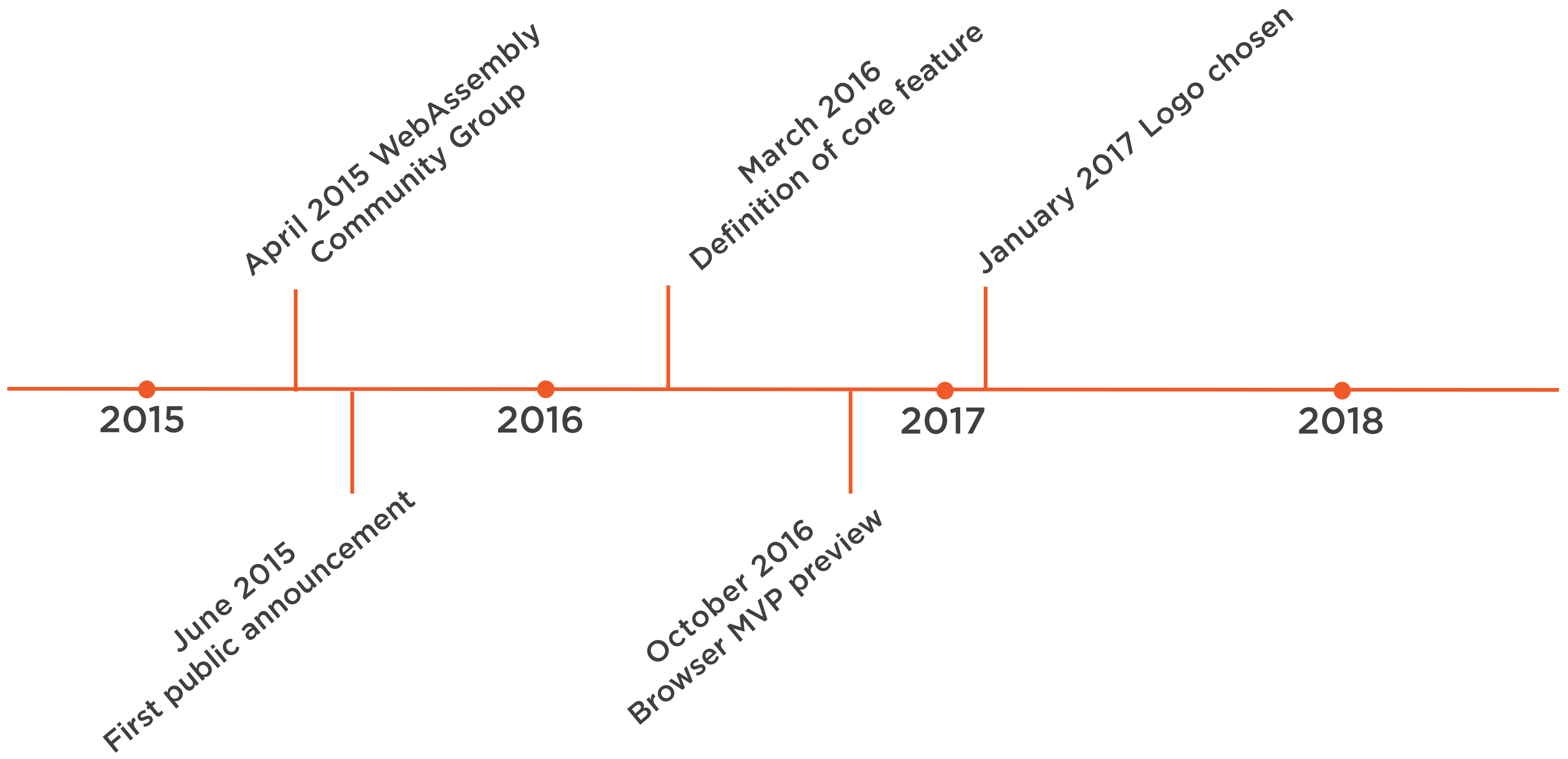
How to work with it

Where Did WebAssembly Come From?

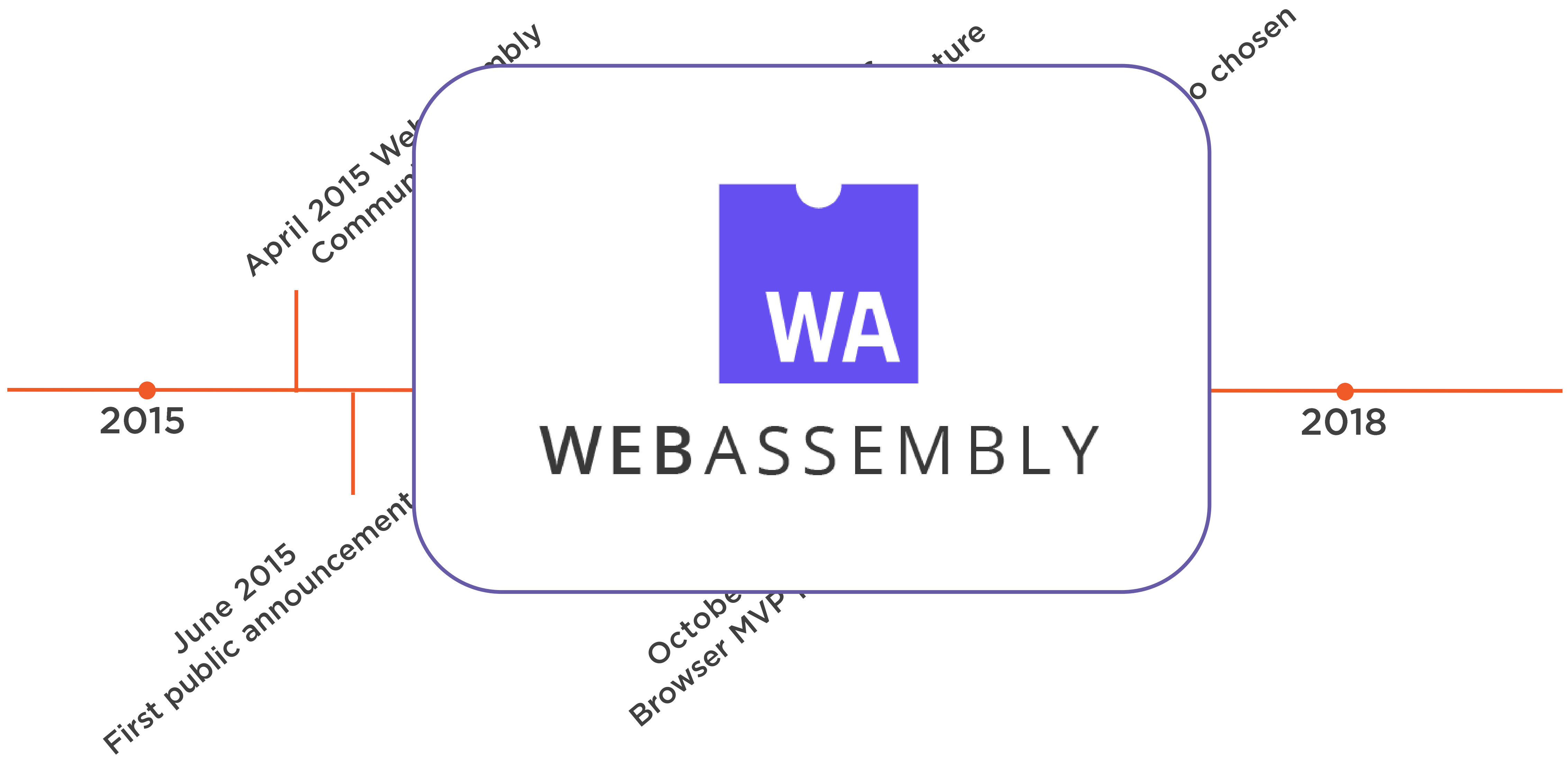
WebAssembly is a new type of code that can be run in modern web browsers and provides new features and major gains in performance.

Source: https://developer.mozilla.org/en-US/docs/WebAssembly/Concepts

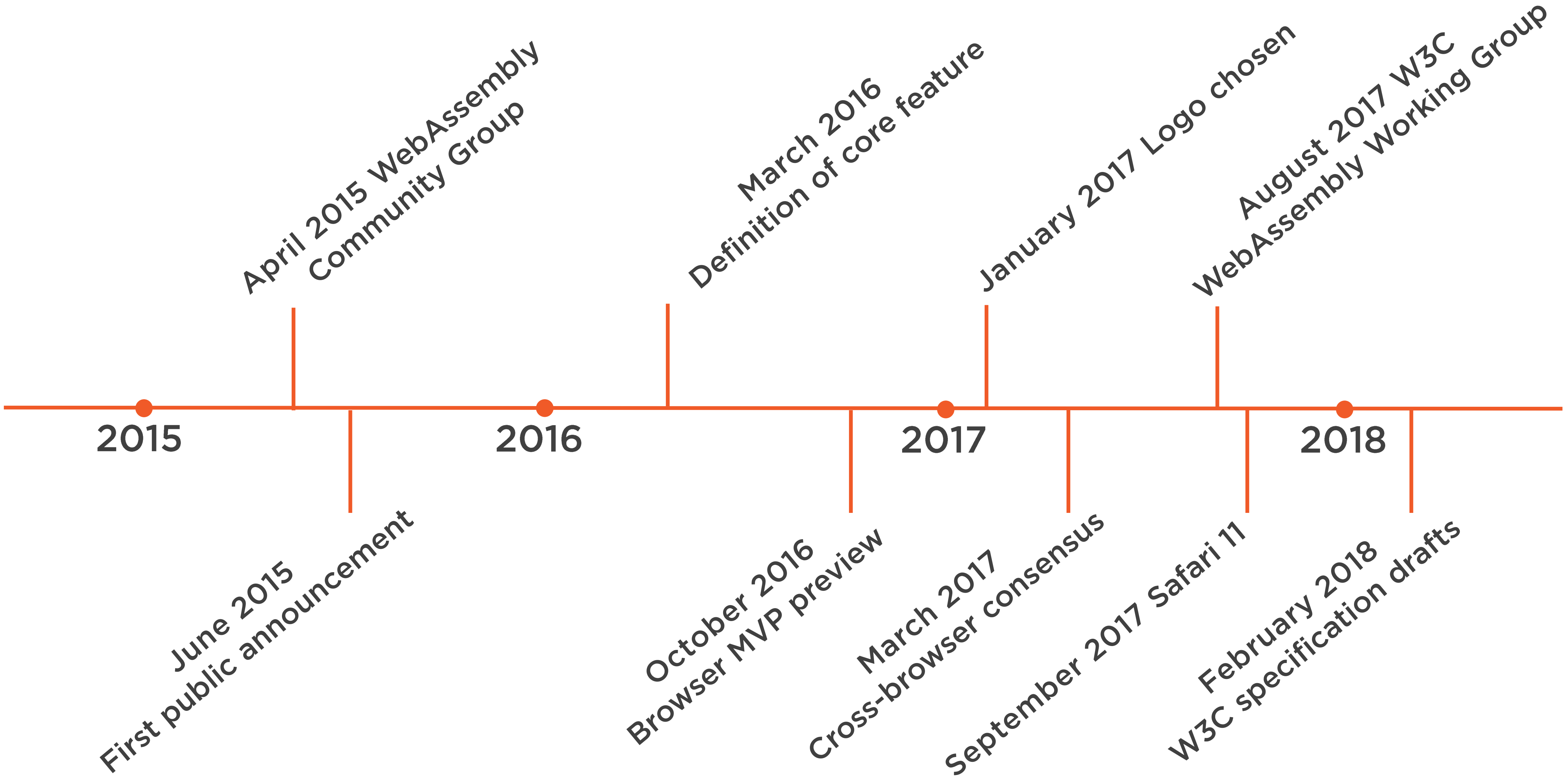
History of WebAssembly



History of WebAssembly



History of WebAssembly

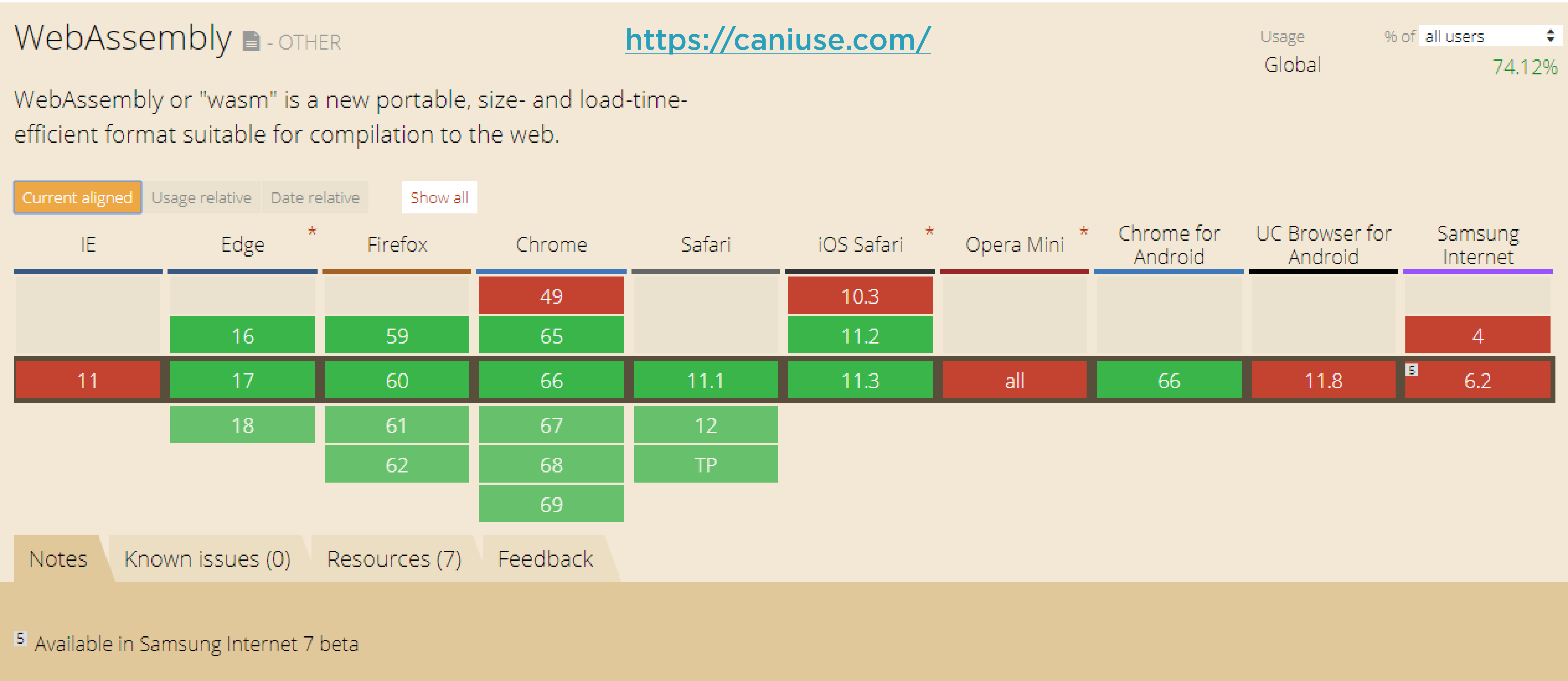

April 2015 WebAssembly Community Group
June 2015 First public announcement
March 2016 Definition of core feature
October 2016 Browser MVP preview
January 2017 Logo chosen
March 2017 Cross-browser consensus
August 2017 W3C WebAssembly Working Group
September 2017 Safari 11
February 2018 W3C specification drafts
2015
2016
2017
2018

Who Is Developing WebAssembly?

- 2015: WebAssembly Community Group
    - Representatives from major browser vendors and community (900+ members)
    - Promote early-stage cross-browser collaboration on WebAssembly

- 2017: W3C WebAssembly Working Group
    - Google, Apple, Facebook, Tencent, W3C, Mozilla, LG Electronics and more
    - Standardize WebAssembly

Which Browsers Support WebAssembly?



WebAssembly - OTHER

https://caniuse.com/

Usage % of all users
Global 74.12%

WebAssembly or "wasm" is a new portable, size- and load-time-efficient format suitable for compilation to the web.

Current aligned | Usage relative | Date relative | Show all

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			🟥 49		🟥 10.3				
	🟩 16	🟩 59	🟩 65		🟩 11.2				🟥 4
🟥 11	🟩 17	🟩 60	🟩 66	🟩 11.1	🟩 11.3	🟥 all	🟩 66	🟥 11.8	🟥 [5] 6.2
	🟩 18	🟩 61	🟩 67	🟩 12					
		🟩 62	🟩 68	🟩 TP					
			🟩 69						

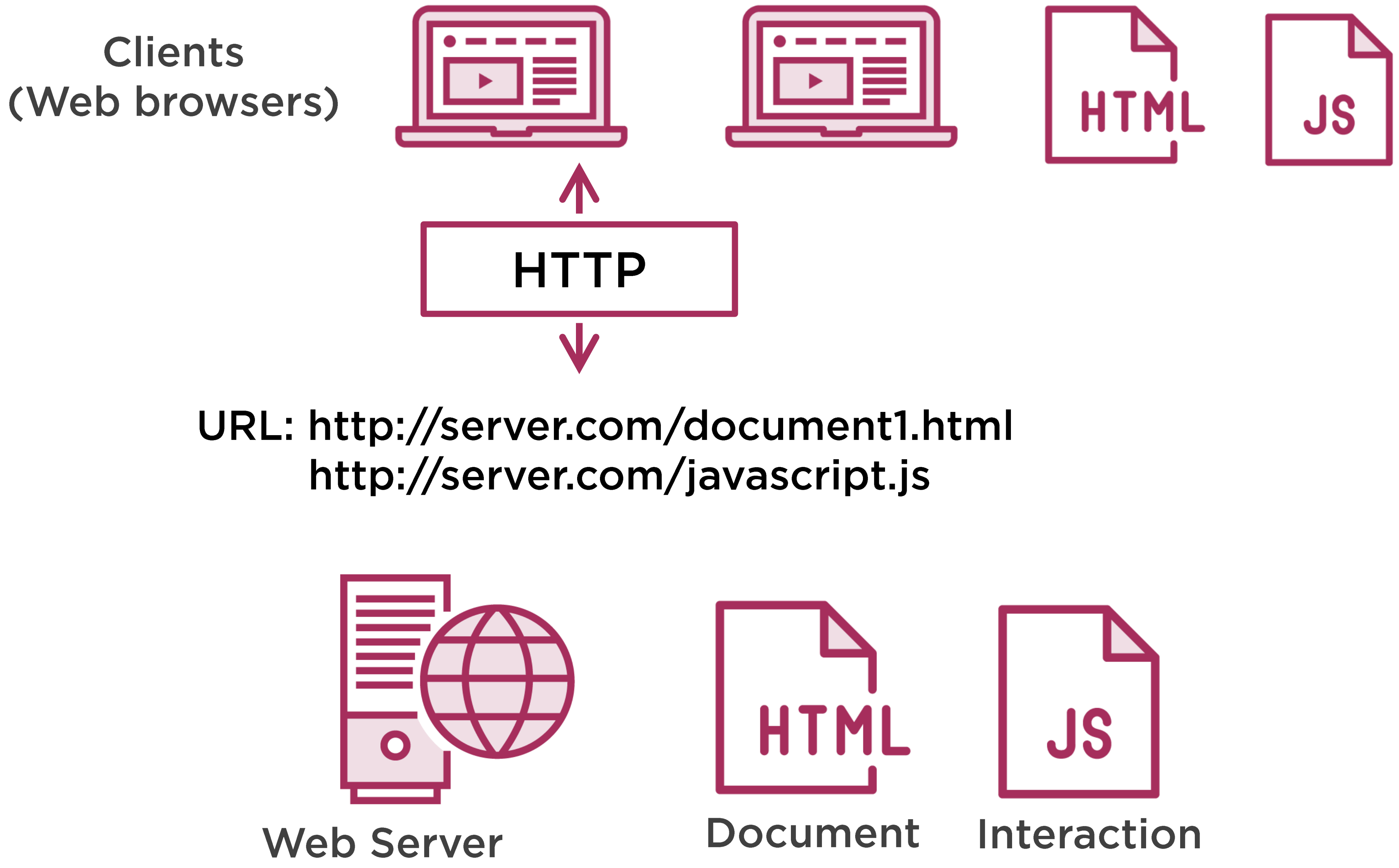
Notes | Known issues (0) | Resources (7) | Feedback

[5] Available in Samsung Internet 7 beta

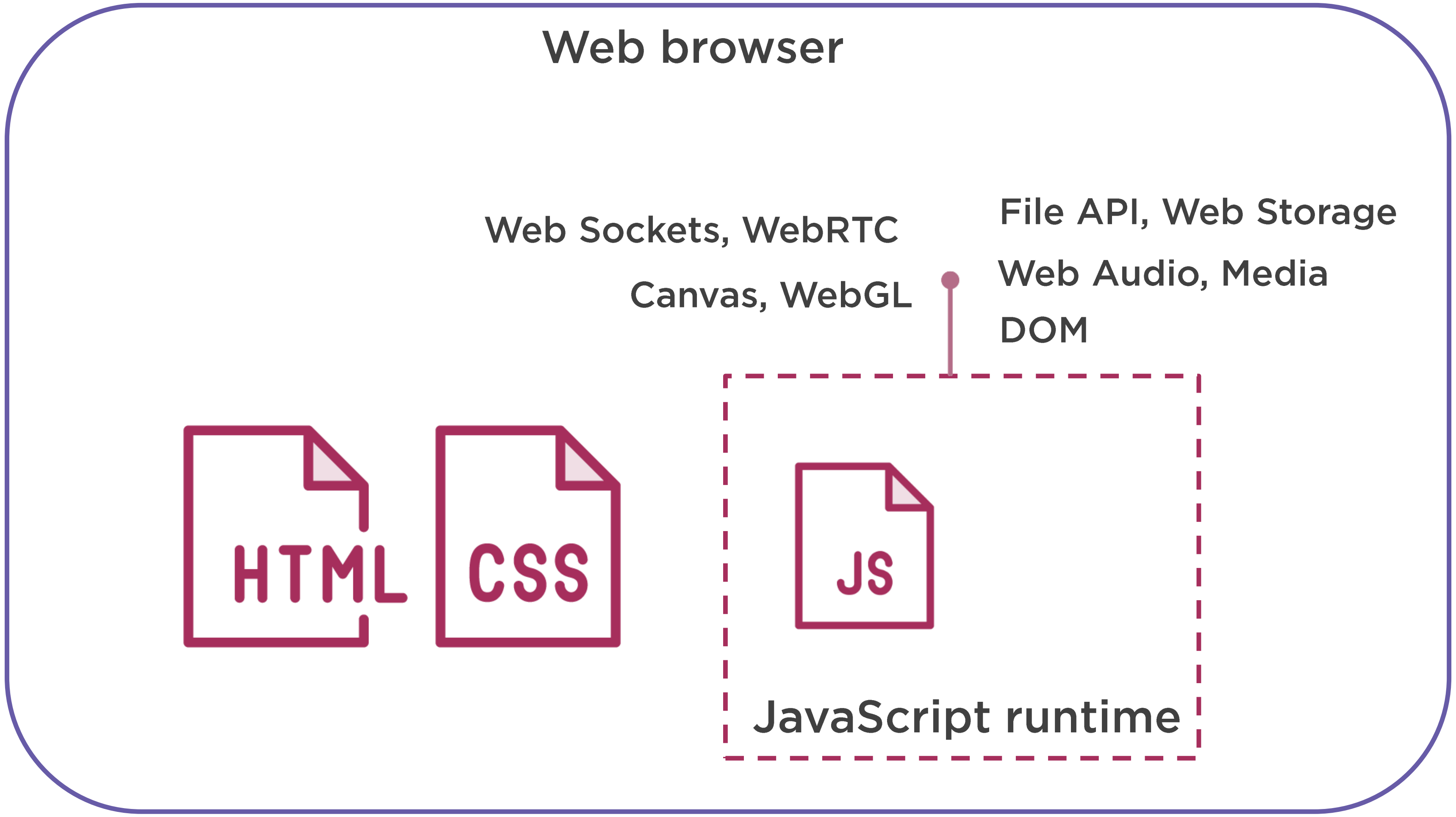
What Is WebAssembly?

JavaScript is a high-level, interpreted programming language

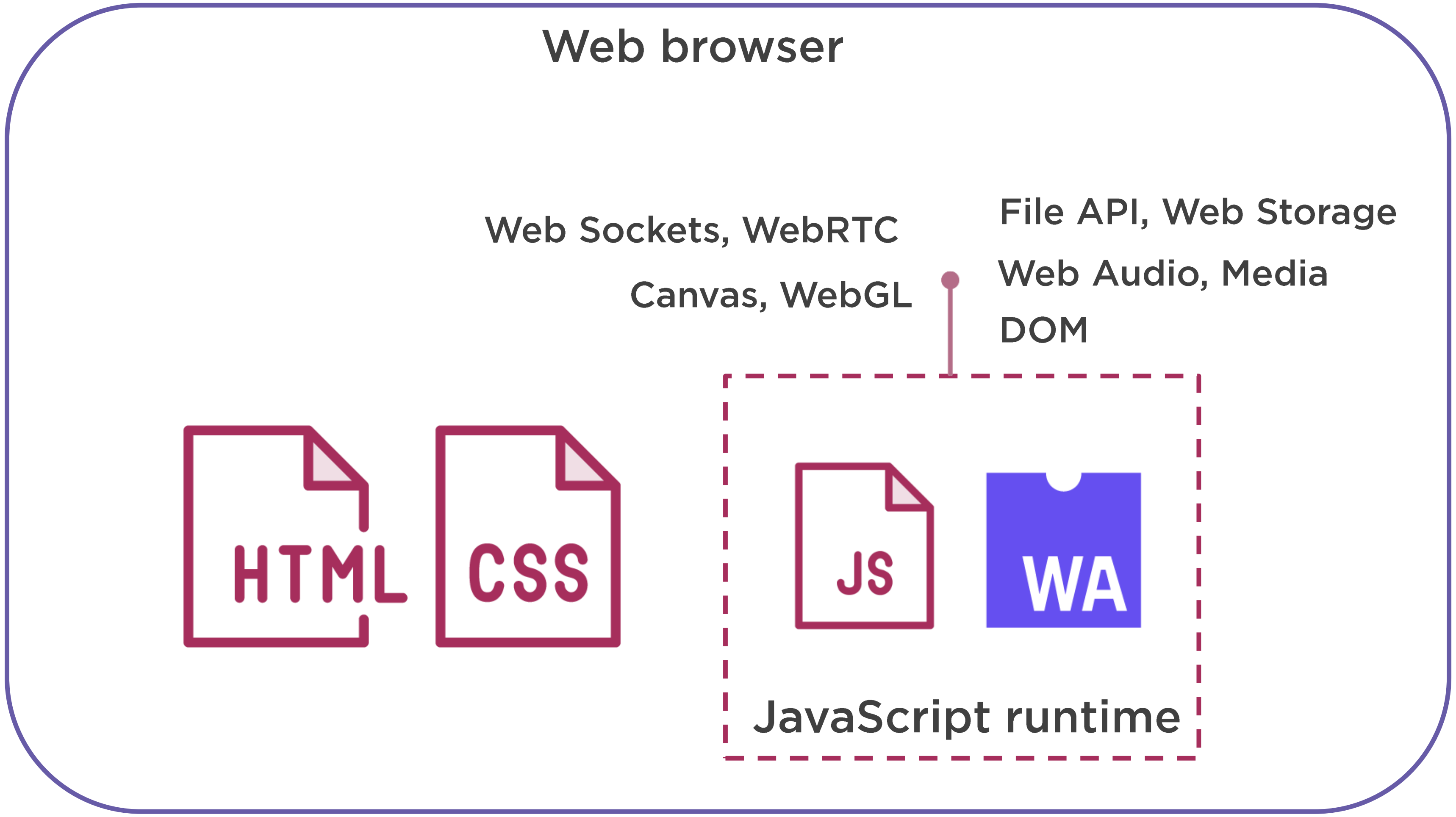
JavaScript in a Nutshell


Clients
(Web browsers)
HTML
JS
HTTP
URL: http://server.com/document1.html
http://server.com/javascript.js
Web Server
HTML
Document
JS
Interaction

JavaScript in a Nutshell



WebAssembly



WebAssembly

JS
WA
JavaScript runtime

WebAssembly

JS
WA
JavaScript runtime

```
function ShowDate() {
   document.getElementById('demo’)
            .innerHTML = Date();
}
```

Main.js

```
0x00000000 0061736D0100000001 .asm..........
0x00000010 7F0302010007070103 .........add....
0x00000020 010700200020016A0B ... . .j....name
0x00000030 010601000361646402 .....add.......l
0x00000040 68730103726873     hs..rhs
```

Main.wasm

WebAssembly

JS
WA
JavaScript runtime

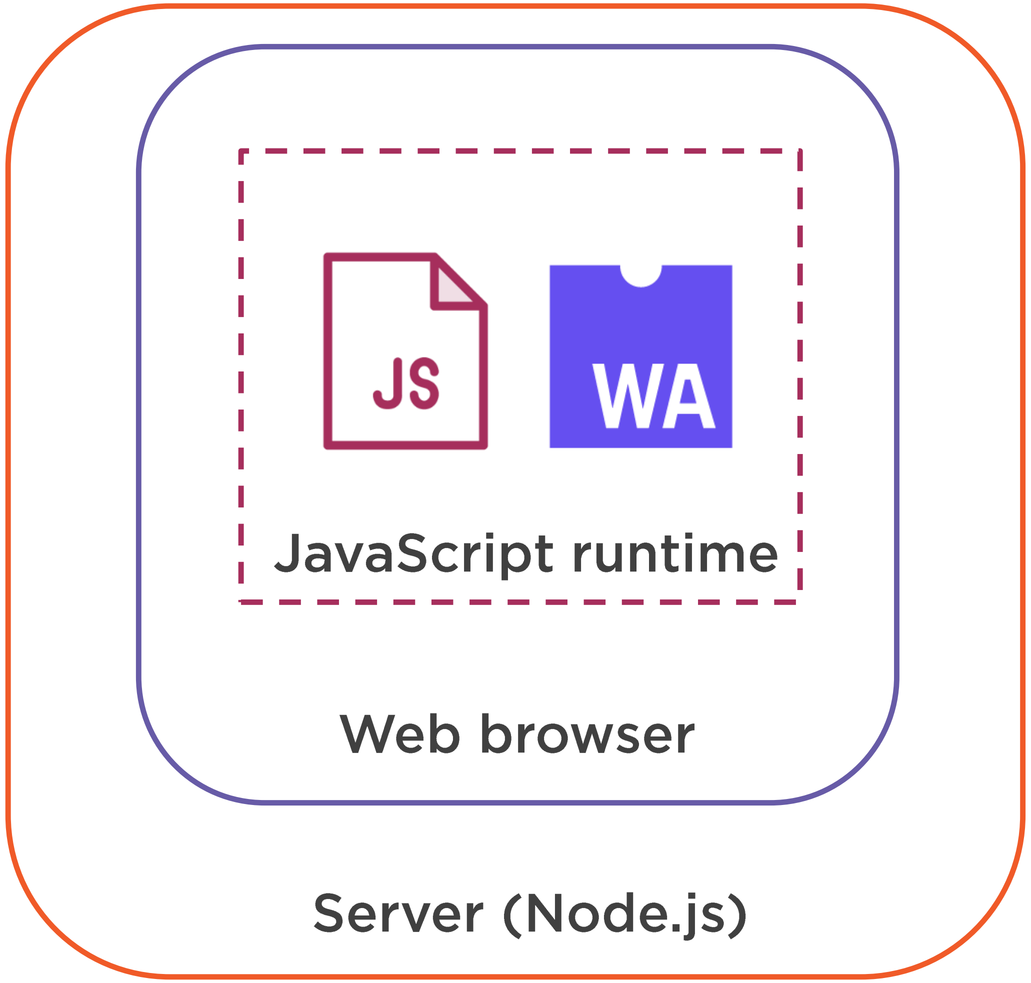
```
function ShowDate() {
   document.getElementById('demo’)
           .innerHTML = Date();
}
```

Main.js

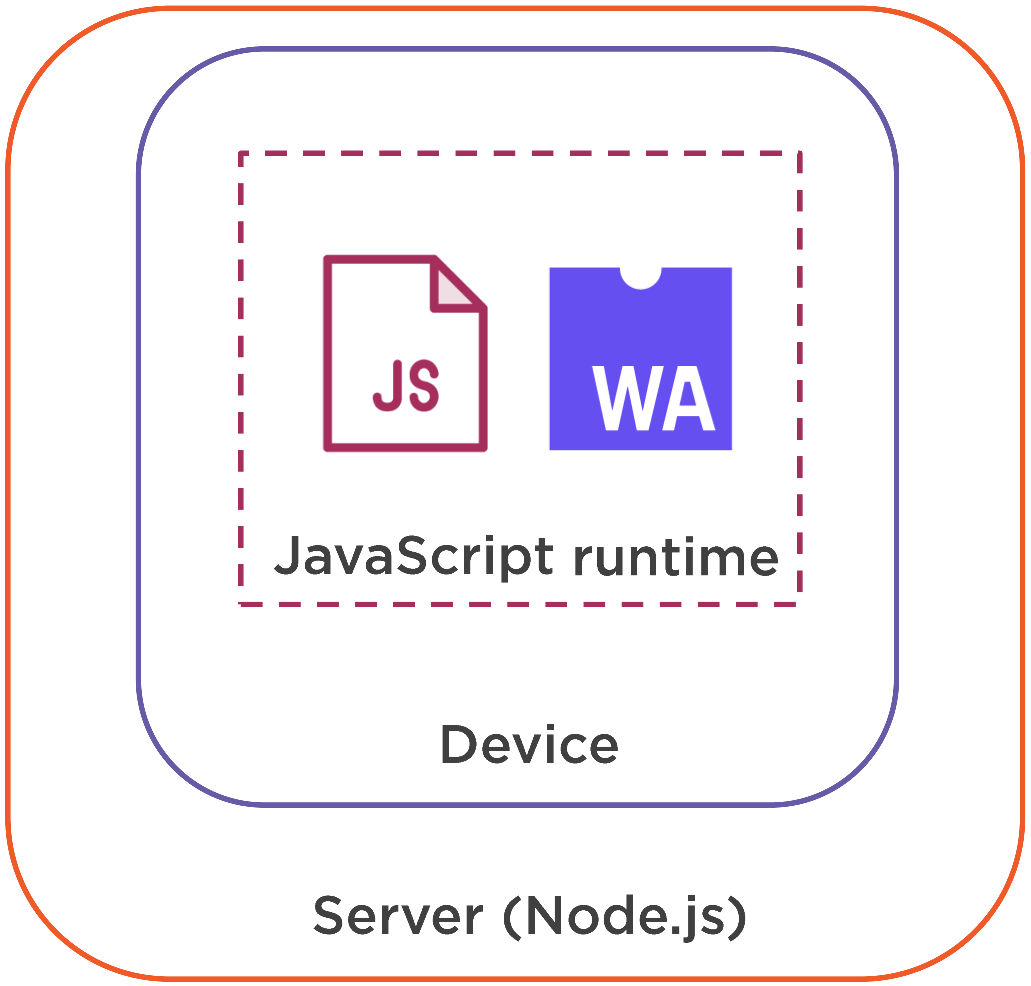
```
0x00000000 0061736D0100000001 .asm...........
0x00000010 7F0302010007070103 .........add....
0x00000020 010700200020016A0B ... . .j....name
0x00000030 010601000361646402 .....add.......l
0x00000040 68730103726873     hs..rhs
```

Main.wasm

WebAssembly



WebAssembly



WebAssembly (abbreviated *Wasm*) is a binary instruction format for a stack-based virtual machine.

Wasm is designed as a portable target for compilation of high-level languages like C/C++/Rust,

enabling deployment on the web for client and server applications.

https://webassembly.org/

Why WebAssembly?

Run code at near-native speed

Other languages can be compiled to WebAssembly

Natively supported by browsers – no plugin needed

Secure by design – it runs in the JavaScript sandbox

JavaScript code can run WebAssembly modules

WebAssembly Use Cases

Video / audio editing

Video / audio streaming

Gaming

Video / audio calling

Virtual / Augmented reality

Artificial Intelligence

WebAssembly Example

https://www.funkykarts.rocks/demo.html

Working with WebAssembly

Ways to Use WebAssembly

Write WebAssembly yourself

Compile code into WebAssembly

Use WebAssembly modules from JavaScript

Write WebAssembly Yourself

The WebAssembly Binary Toolkit → Convert → WA runtime

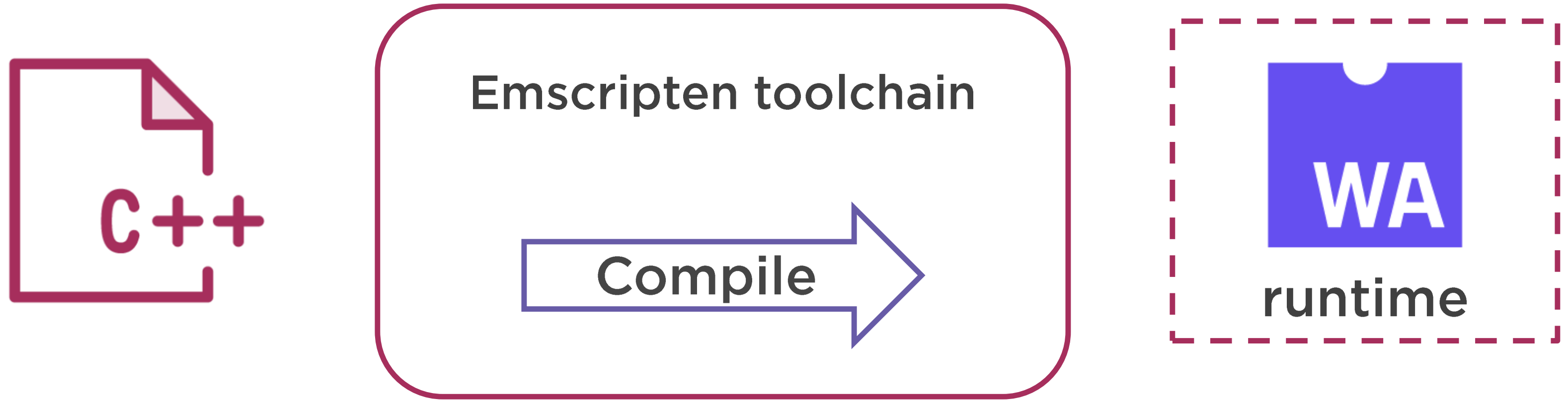
```
(module
  (func $add (param $lhs i32)
  (param $rhs i32) (result i32)
    get_local $lhs
    get_local $rhs
    i32.add)
  (export "add" (func $add))
)
```

Main.wat

```
0x00000000 0061736D0100000001 .asm..........
0x00000010 7F0302010007070103 .........add....
0x00000020 010700200020016A0B ... . .j....name
0x00000030 010601000361646402 .....add.......l
0x00000040 68730103726873     hs..rhs
```

Main.wasm

Compile Code into WebAssembly



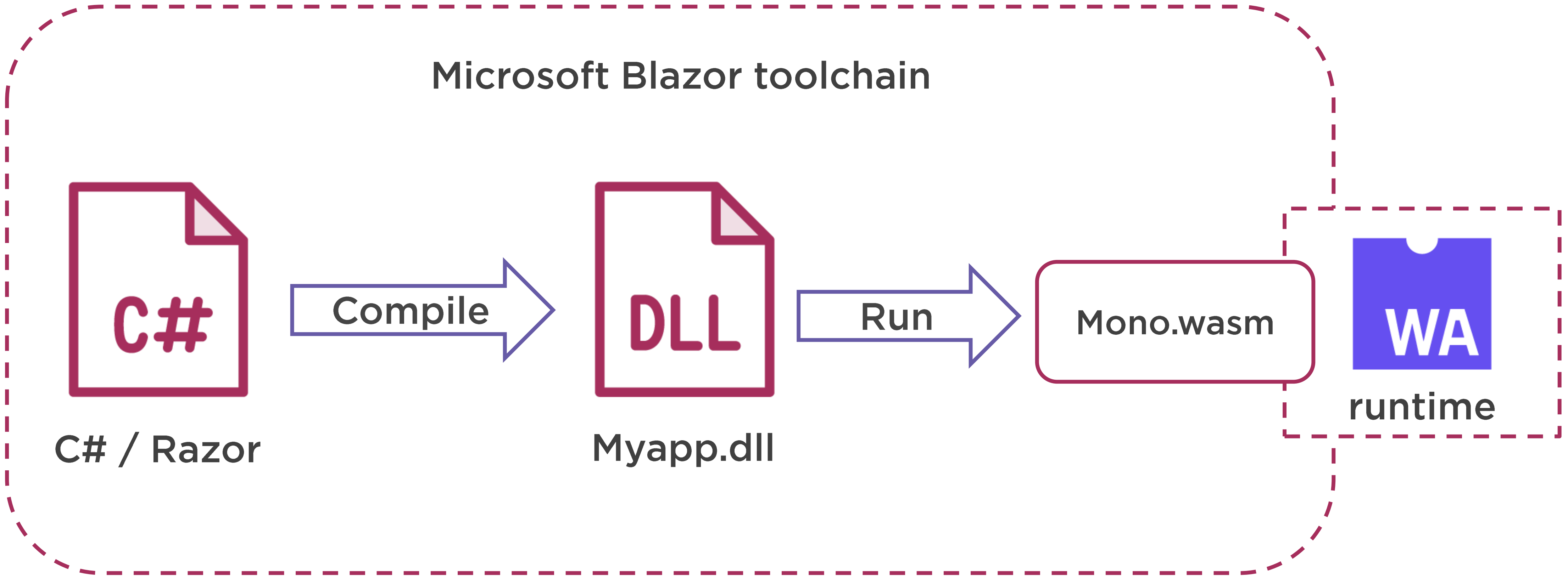
```cpp
#include <iostream>

int main()
{
    std::cout << "Hello, World!";
    return 0;
}
```

MyCApp.cpp

```
0x00000000 0061736D0100000001 .asm...........
0x00000010 7F0302010007070103 .........add....
0x00000020 010700200020016A0B ... . .j....name
0x00000030 010601000361646402 .....add.......l
0x00000040 68730103726873     hs..rhs
```

MyCApp.wasm

Compile Code into WebAssembly



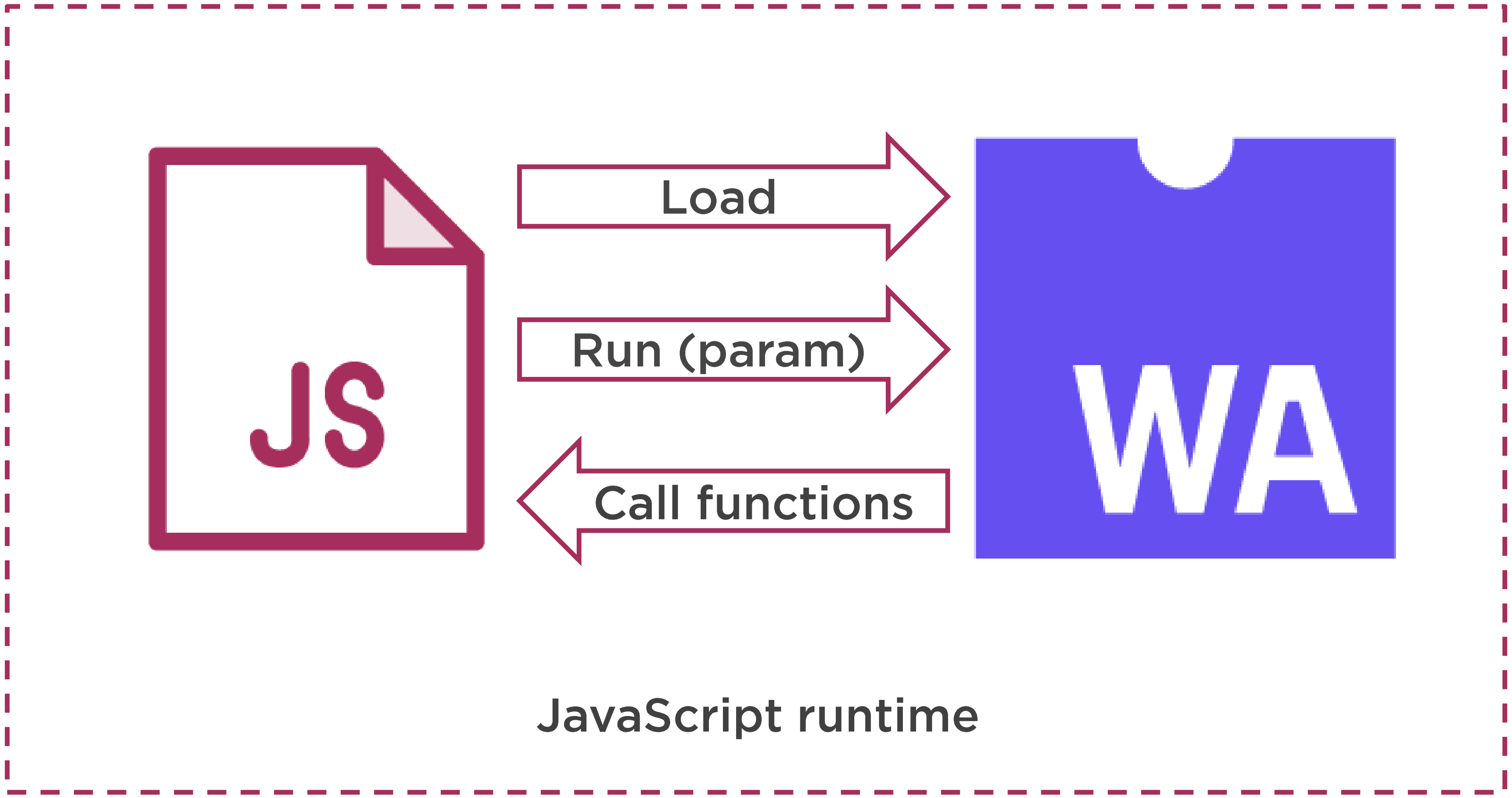
Load WebAssembly Modules in JavaScript

wasm module **wasm module** **wasm module**

```
fetch('../out/main.wasm').then(response =>
  response.arrayBuffer()
).then(bytes => WebAssembly.instantiate(bytes)).then(results => {
  instance = results.instance;
  document.getElementById("container").innerText = instance.exports.add(1,1);
}).catch(console.error);
```

main.js

JavaScript Works with WebAssembly



Things to Remember

WebAssembly is a binary code format

- Created by WebAssembly Community Group
- Being standardized by W3C group

The main benefits of WebAssembly are

- Other languages can be compiled to WebAssembly
- Very performant
- Supported by all the major browsers
  - Uses the JavaScript runtime, without plugins

Some of the use cases are

- Games, VR / AR and streaming

Use WebAssembly by

- Writing it from scratch
- Compiling code into WebAssembly
- Using premade wasm modules