



&

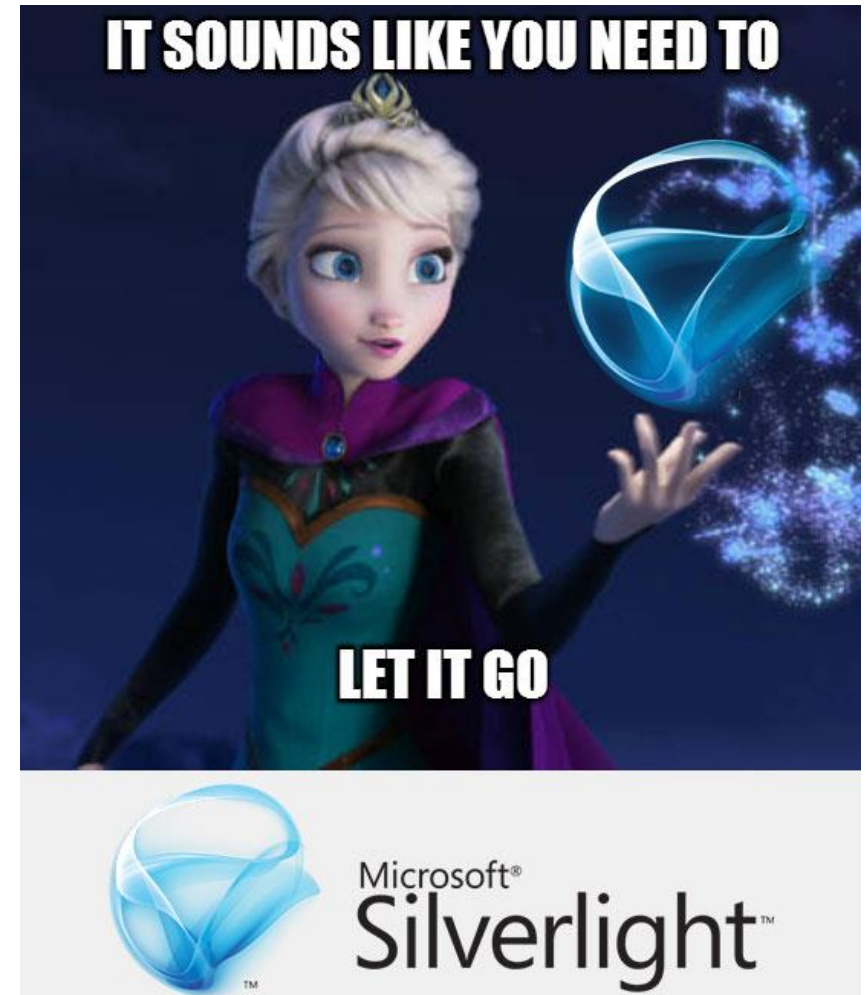


**blazor**

Look ma, no JavaScript... it's Blazor!

# Blazor..

- SPA framework.
- .NET Core 3.0 in the browser.
- Yes, this means No JavaScript.
- Base technology is WebAssembly.
- Near native speeds.
- Open Source.
- Looks like Microsoft has learned from previous experiences..





# WEBASSEMBLY

## Webassembly.org

“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.”

# Short overview

- A low-level binary format, assembly like, language.
- Performance close to native.
- Compatible with 88.8% of browsers.
- A Standard.



«WebAssembly, or wasm, is the most significant new technology to come to the web platform in a decade.»

“it’s the ultimate culmination of web development capabilities”

“WebAssembly is a game-changing technology”

*Source – Mozilla (and many more..)*

<https://research.mozilla.org/webassembly/>

# The story of WebAssembly

Asm.js (2014)

Ext. features  
2018

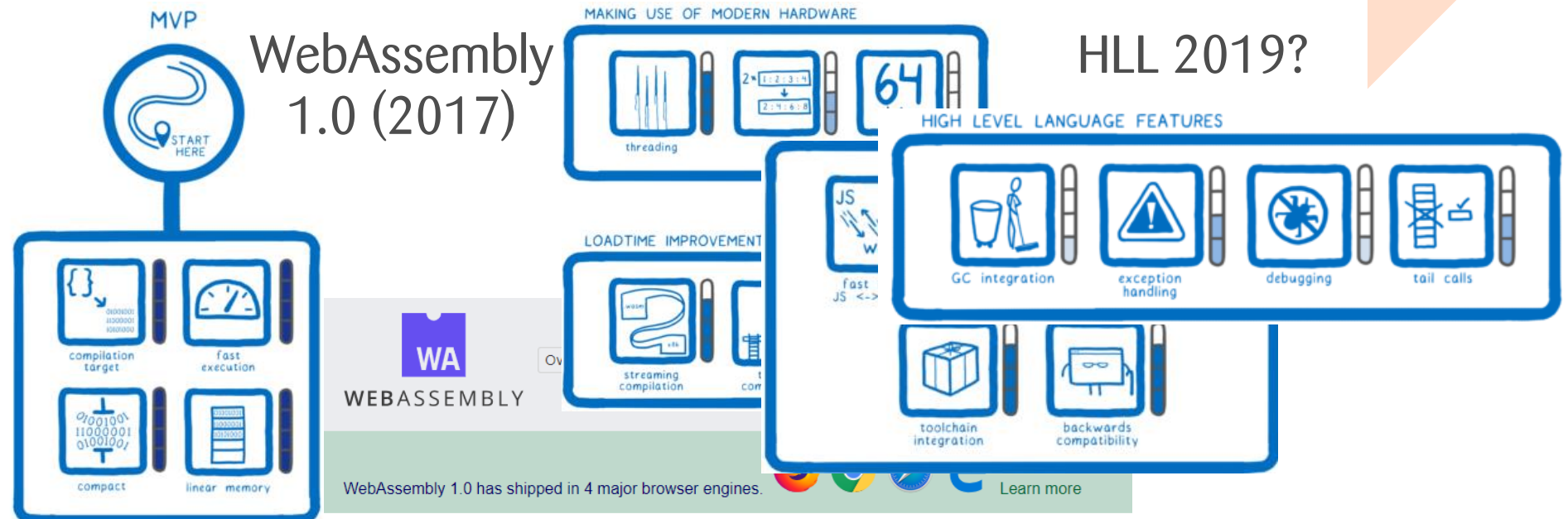
**asm.js**  
Working Draft — 18 August 20

Latest version:  
<http://asmjs.org/spec/latest/>

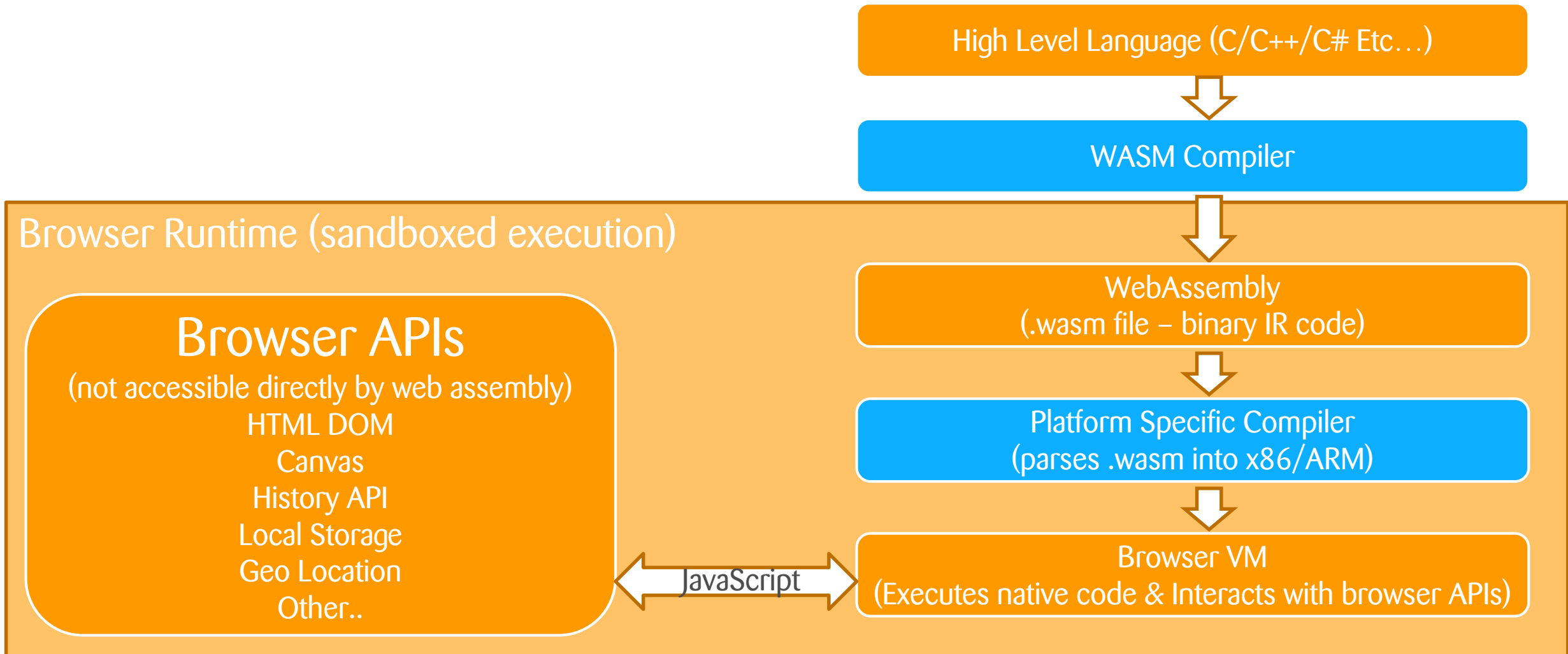
Editors:  
David Herman, Mozilla, <dherman@mozilla.cc>  
Luke Wagner, Mozilla, <luke@mozilla.com>  
Alon Zakai, Mozilla, <azakai@mozilla.com>

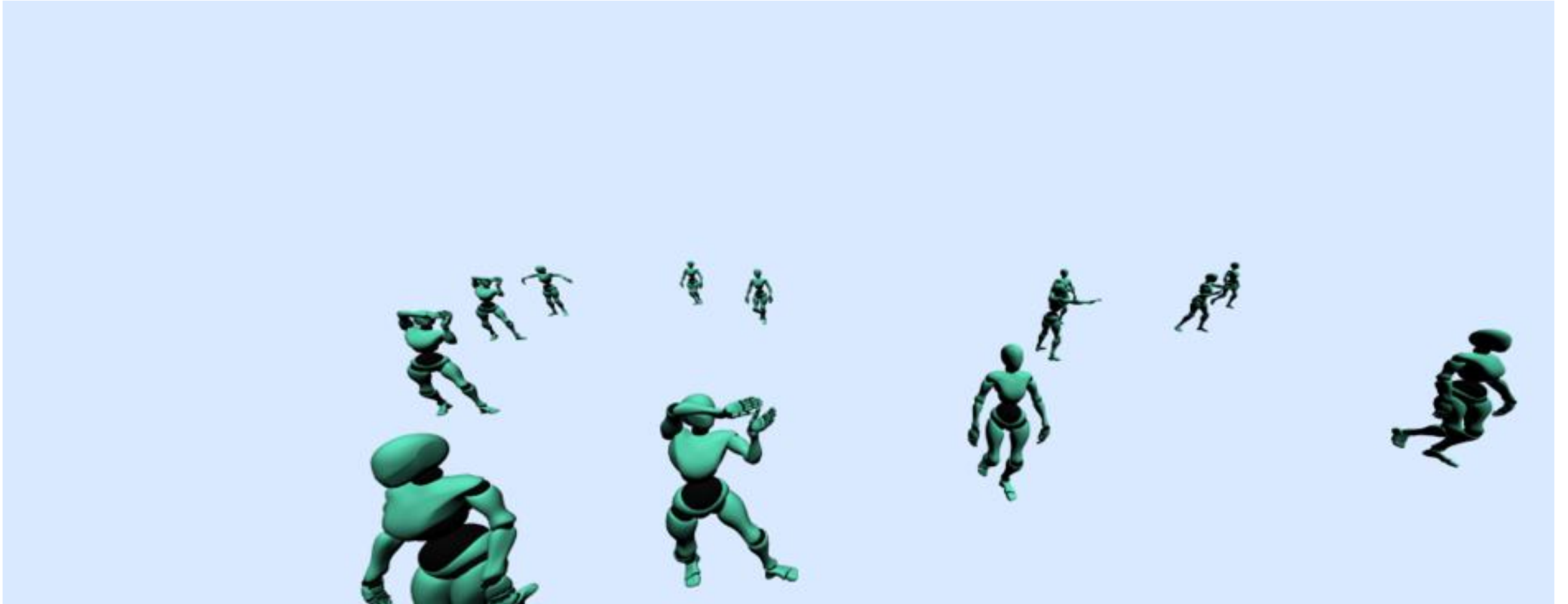
## Abstract

This specification defines **asm.js**, a strict subset of a target language for compilers. This sublanguage eff memory-unsafe languages like C or C++. A combin JavaScript engines to employ an ahead-of-time (AO code.



# WebAssembly simplified





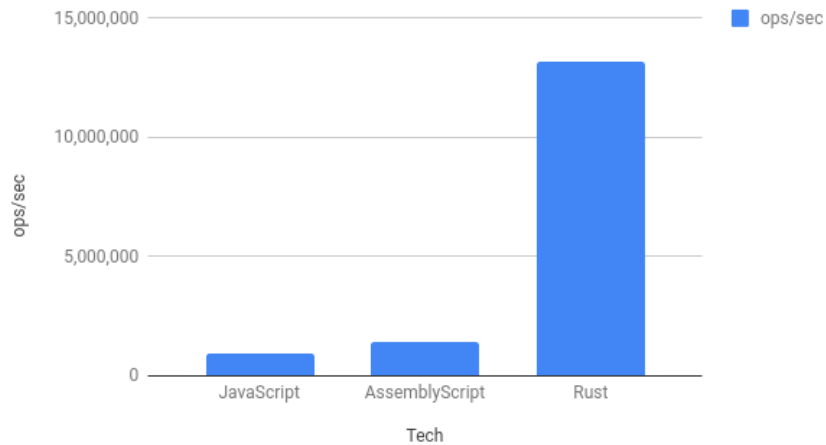
<http://aws-website-webassemblyskeletalanimation-ffaza.s3-website-us-east-1.amazonaws.com/>

# DEMO

Doom3 (nsfw) - <https://wasm.continuation-labs.com/d3demo/>

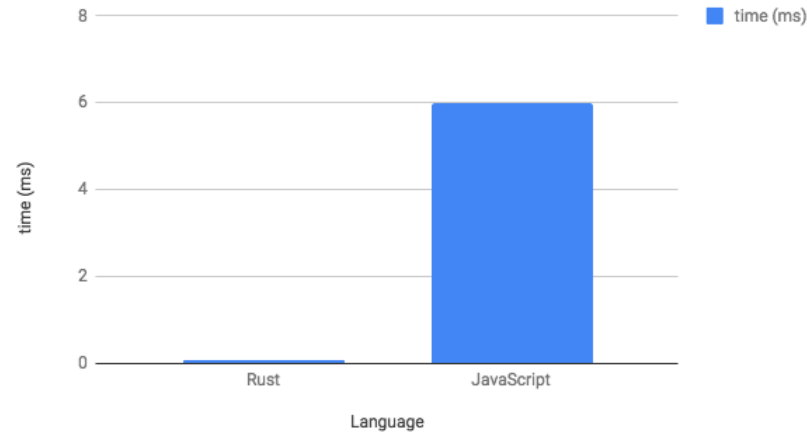
# Benchmarks?

ops/sec (higher is better)



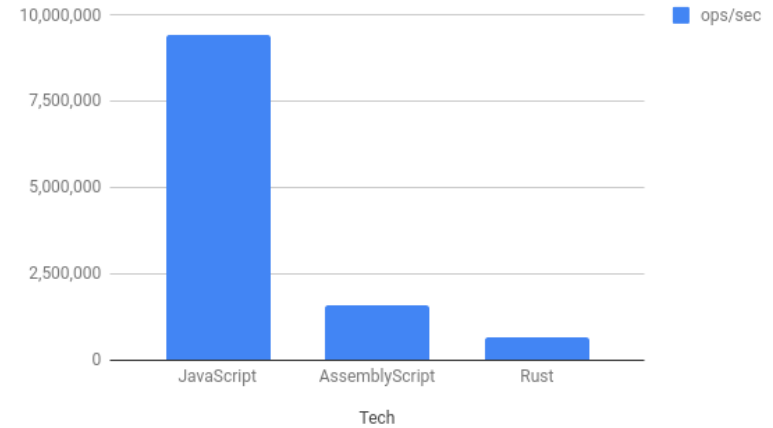
addition

time (ms) (smaller is better)



Numeric Array handling

ps/sec (higher is better)

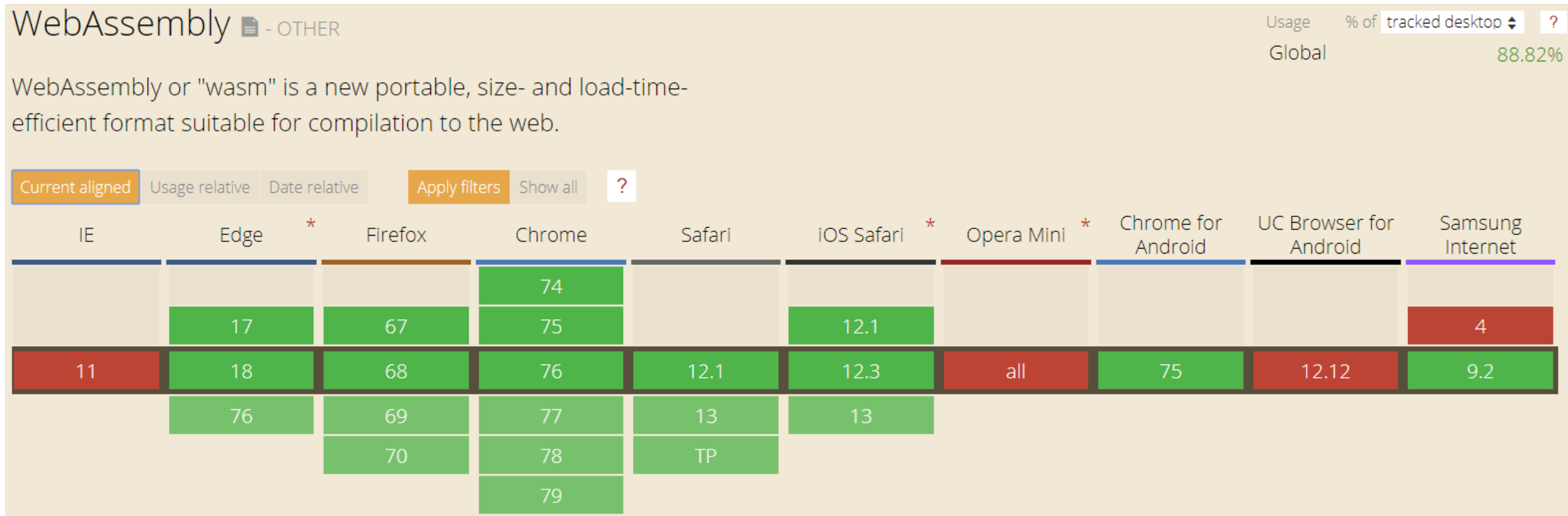


String manipulation

Source: <https://blog.sqreen.com/webassembly-performance/>



# Adoption



Source: <https://caniuse.com/#feat=wasm>



&   
blazor

# Why.NET for browser apps?

Modern  
languages (C#)

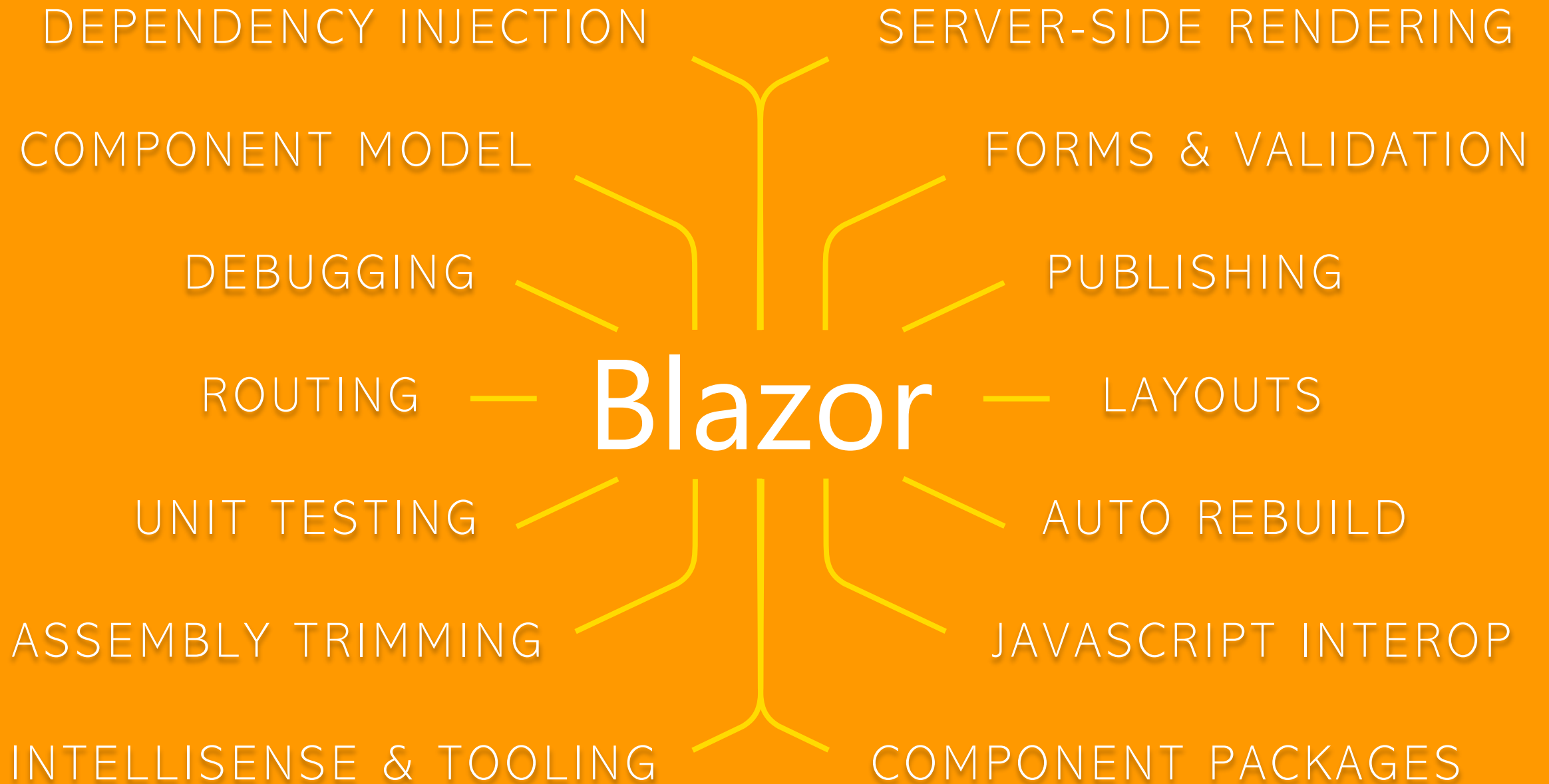
.NET  
ecosystem

Performance

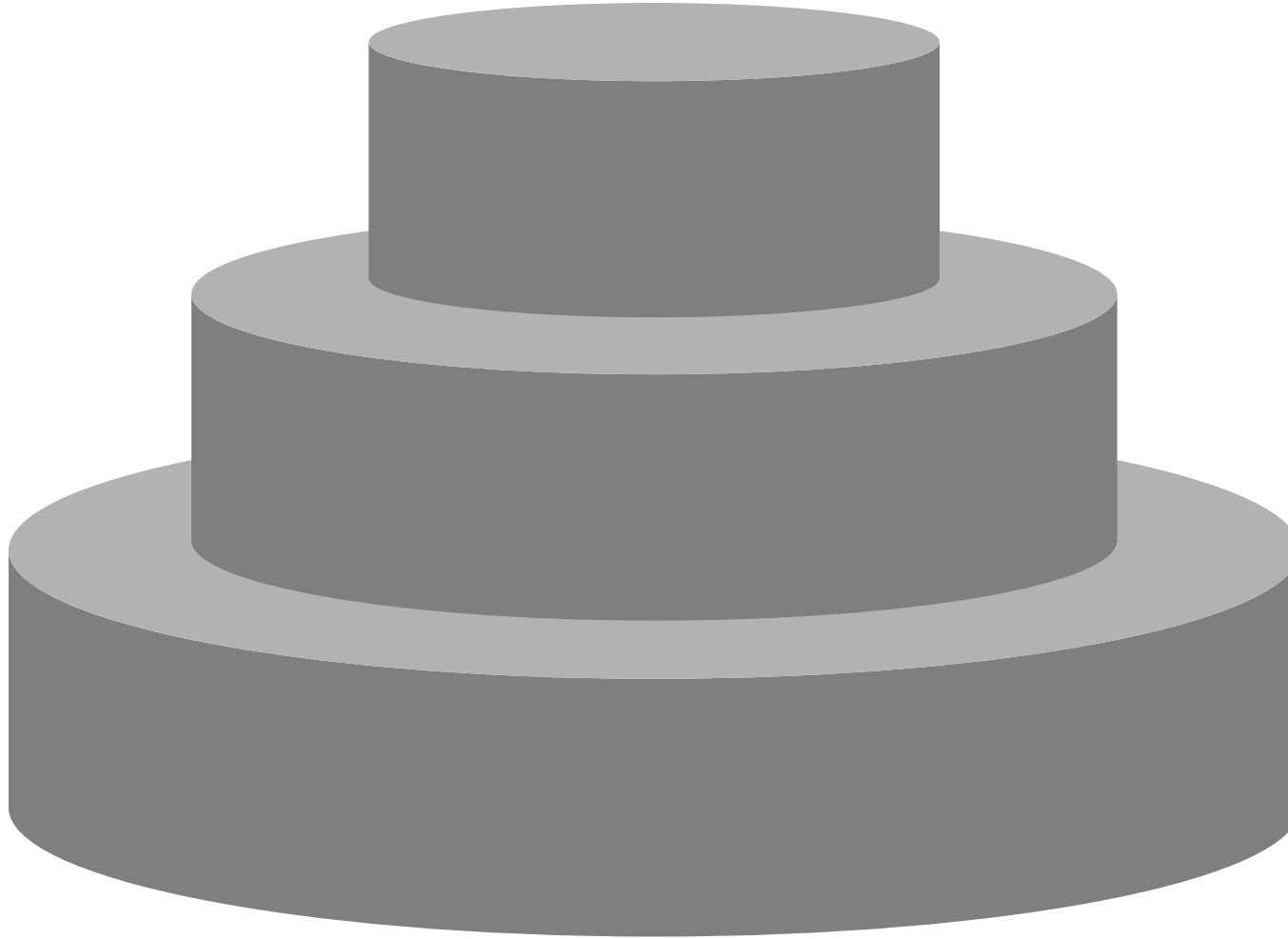
Full-stack

Tools

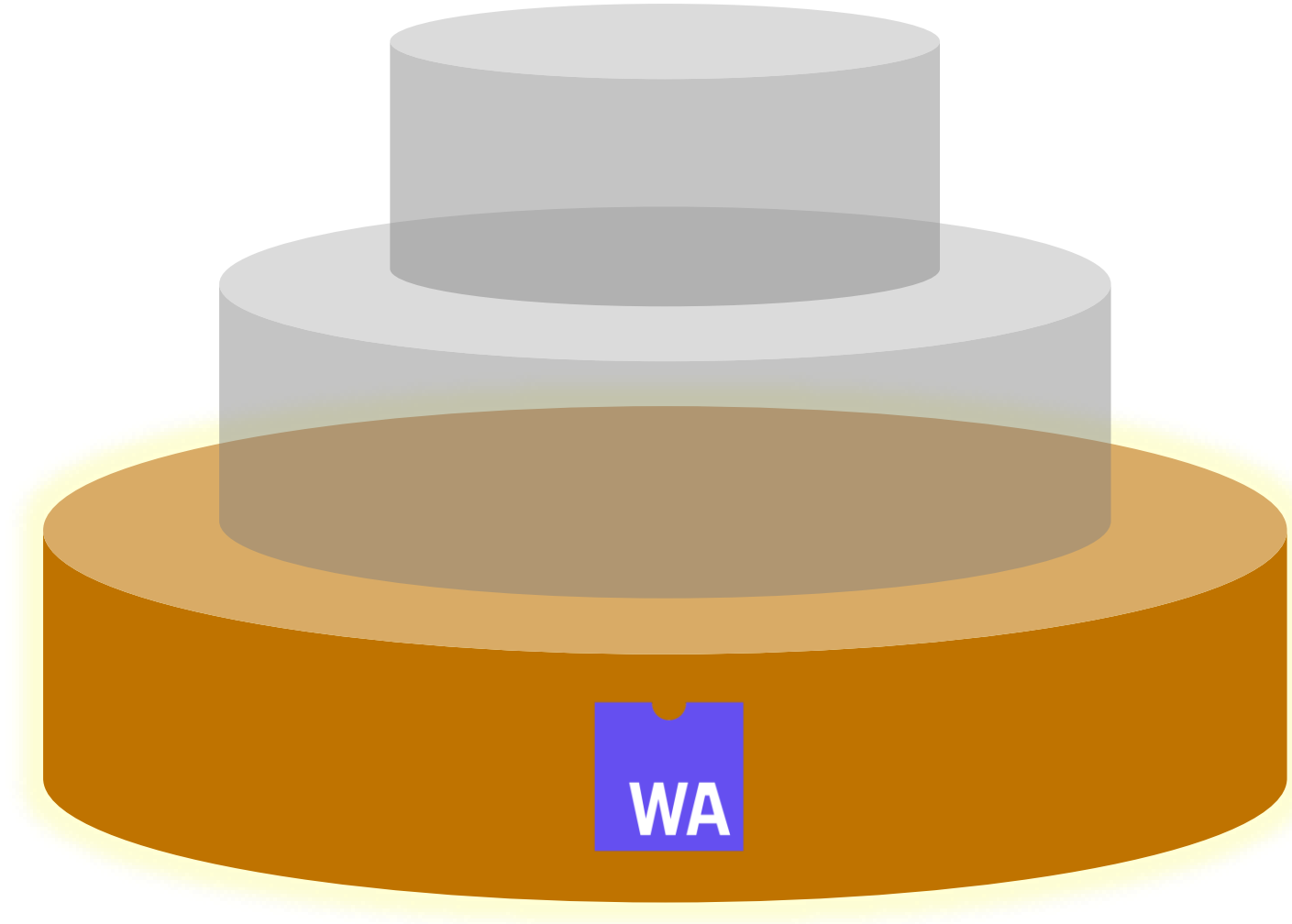
Stable &  
mature



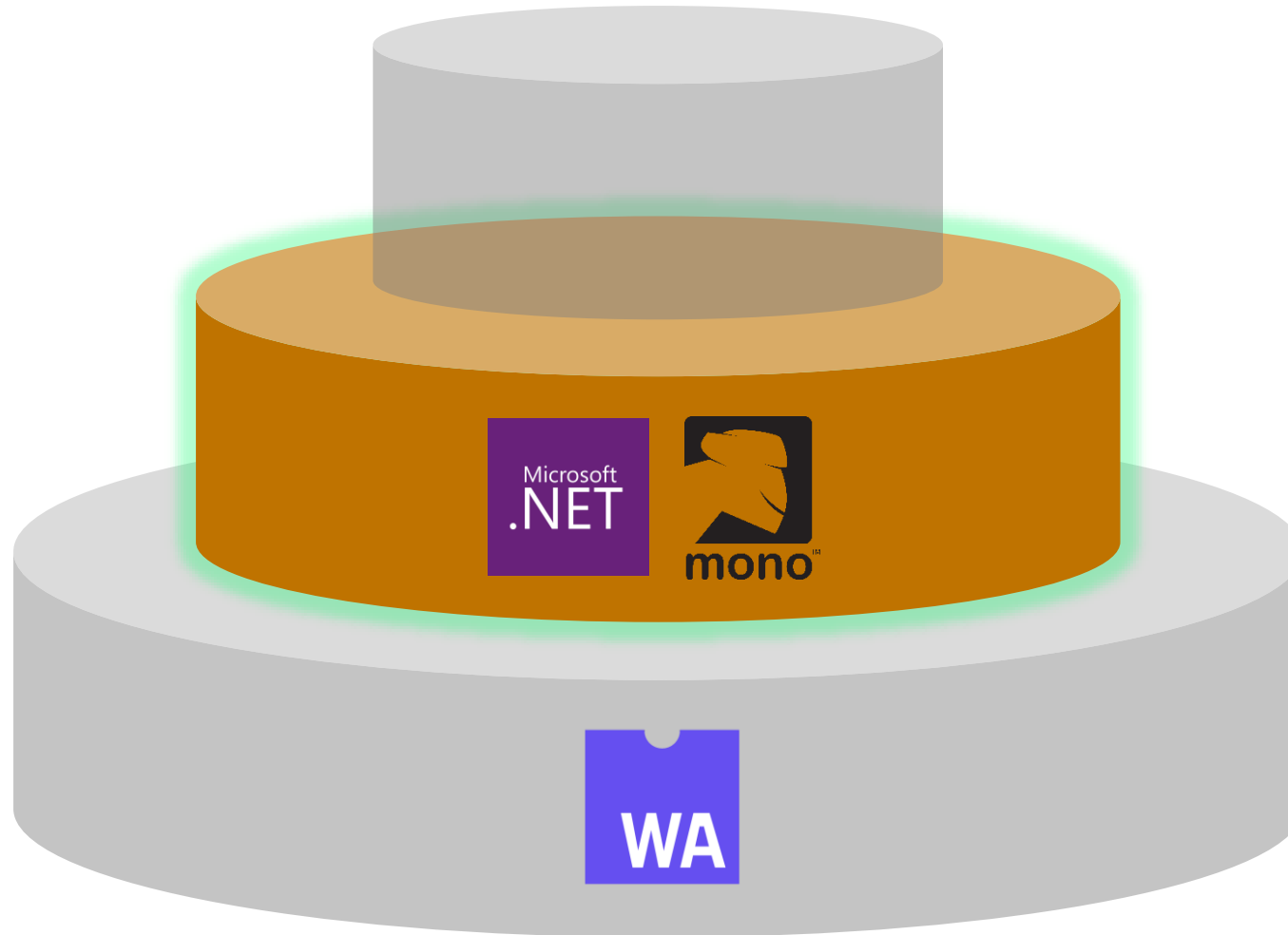
# Ultra simple Architecture



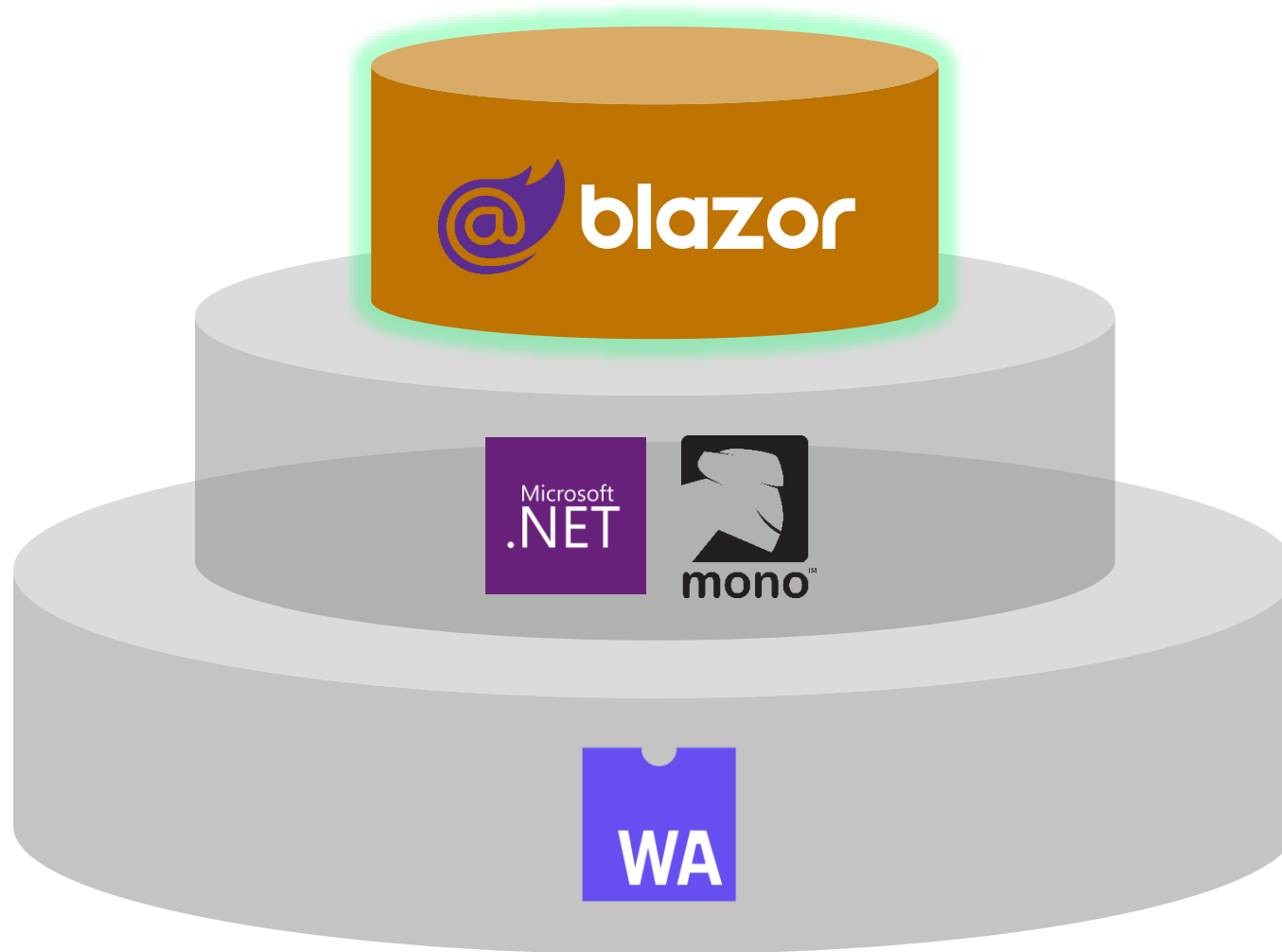
# Ultra simple Architecture



# Ultra simple Architecture

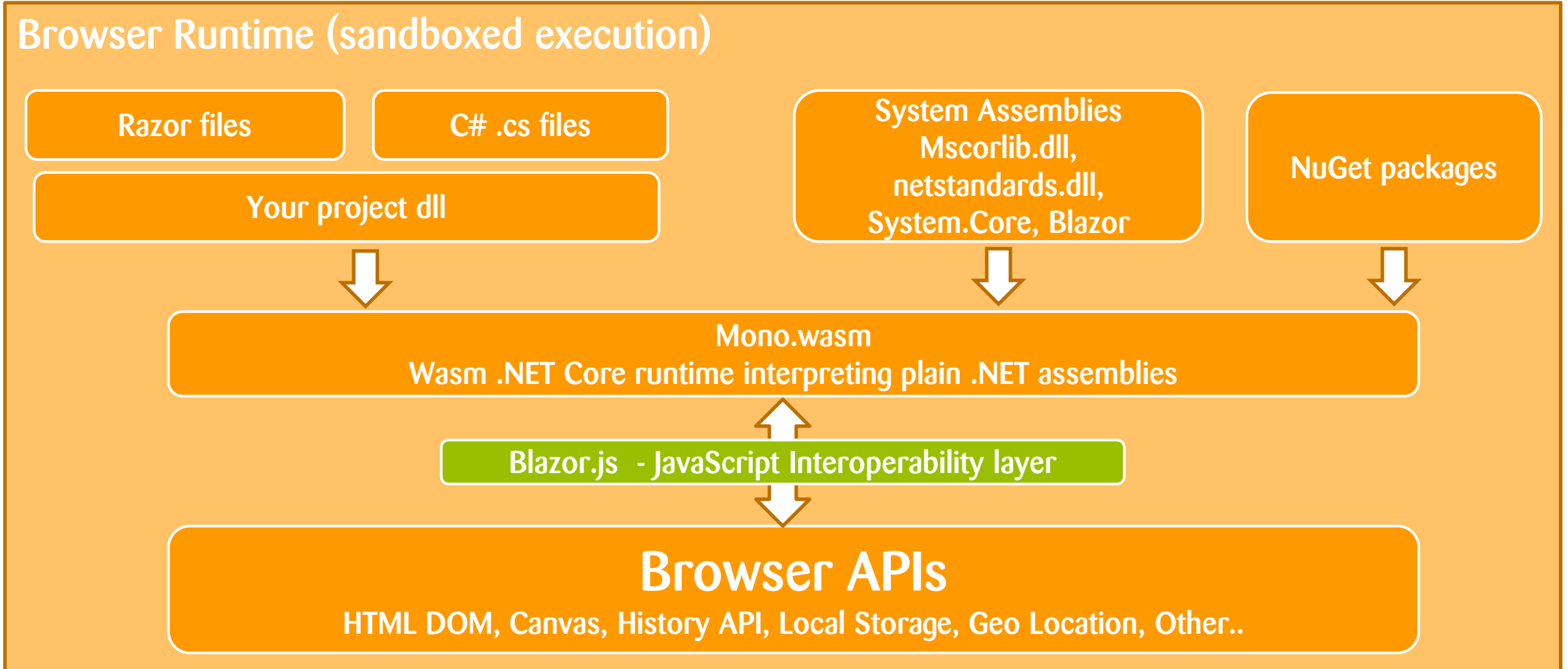


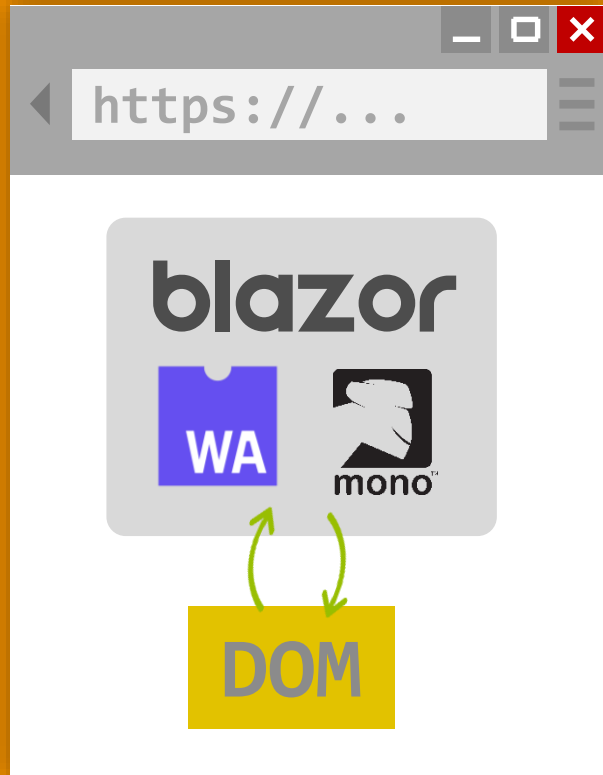
# Ultra simple Architecture



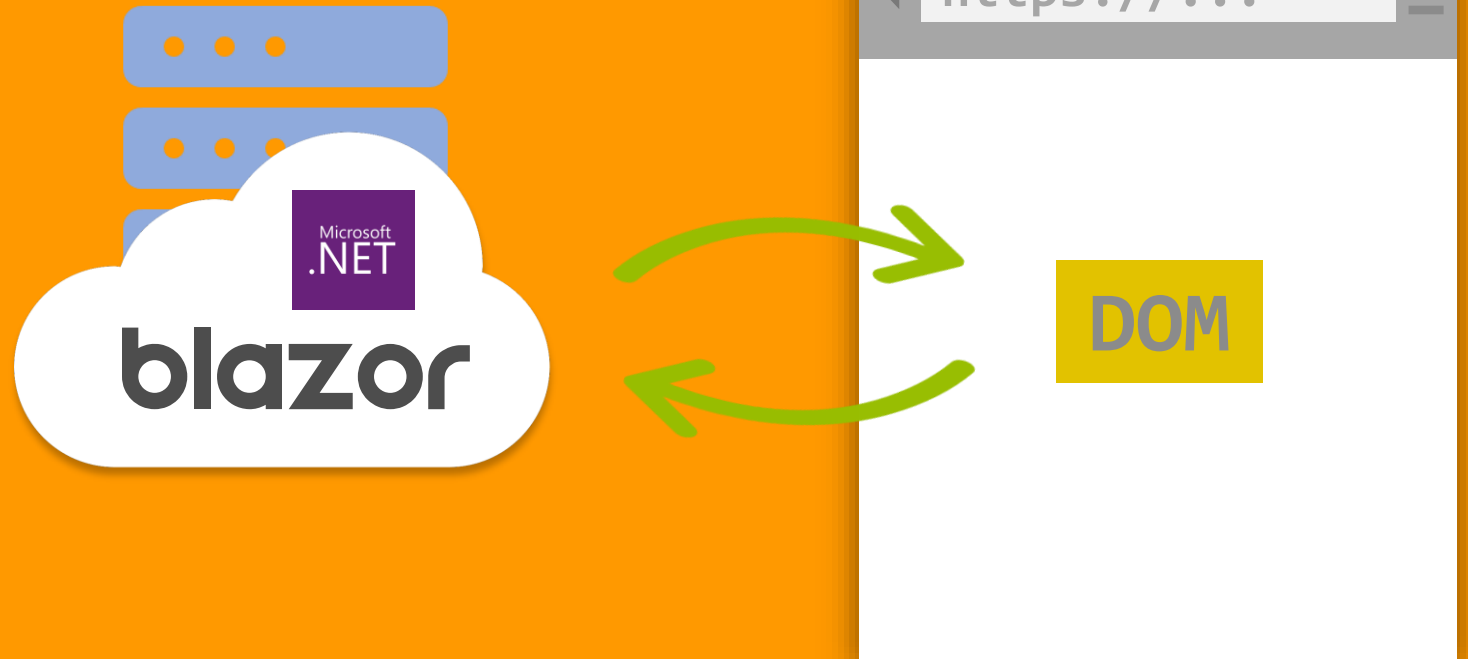


# Blazor simplified

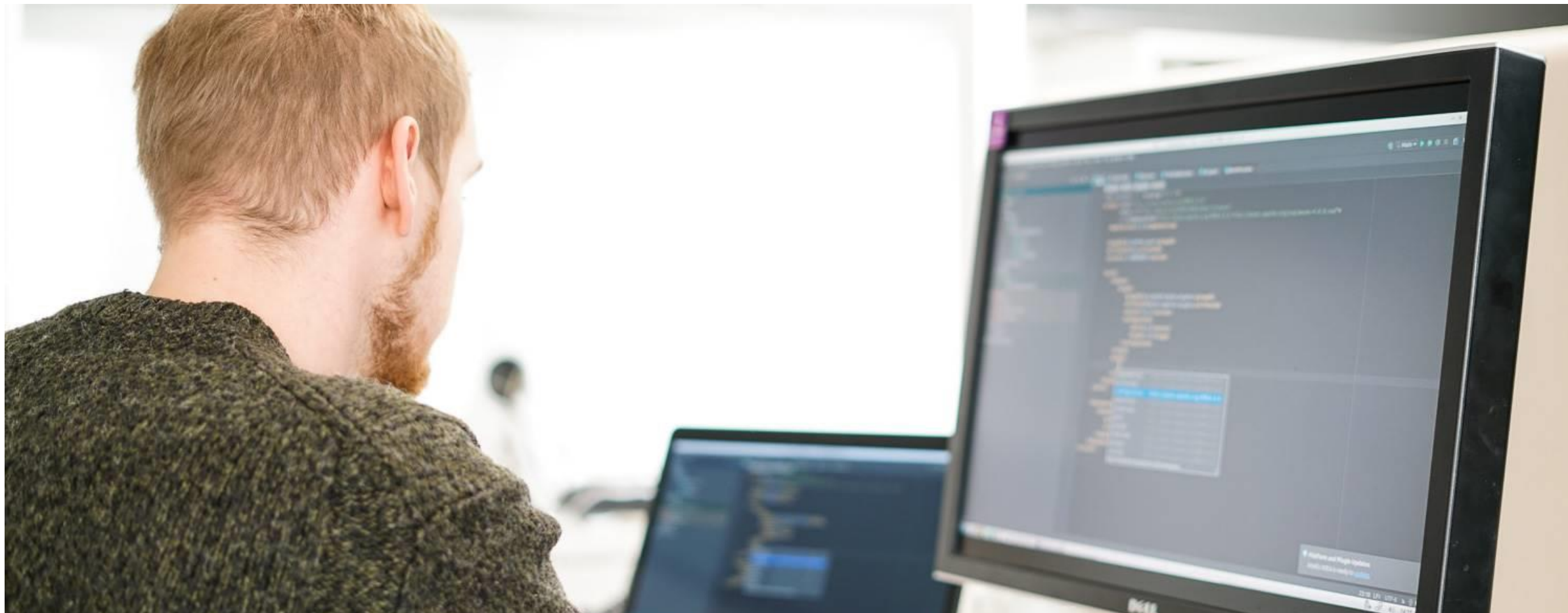




## 1. Client-side



## 2. Server-side



# blazor DEMO(s)

Using Visual Studio & Azure? Take a quick, 5 minute survey to help us make Visual Studio and its Azure-related experiences better.

# Blazor

Build client web apps with C#

Get Started

Documentation

Supported on Windows, Linux, and macOS

```
<h1>Counter</h1>
```

Interactive web UI with C#

# Getting started... With the latest preview

# Go to blazor.net

And follow the yellow path...


- Hint: It will bring you to..
  - <https://dotnet.microsoft.com/apps/aspnet/web-apps/blazor>
- Click on “Get Started” and now, follow the steps..
- Tips
  - Install and use Visual Studio preview (recommended)
  - Otherwise you’ll have issues...


# Solving the issue..

If you do this.. In Visual Studio 2019 latest version...

## Create a new Blazor app

ASP.NET Core 3.1

**Blazor Server App**  
A project template for creating a Blazor server app that runs server-side inside an ASP.NET Core app and handles user interactions over a SignalR connection. This template can be used for web apps with rich dynamic user interfaces (UIs).

**Blazor WebAssembly App**  
A project template for creating a Blazor app that runs on WebAssembly. This template can be used for web apps with rich dynamic user interfaces (UIs).

[Get additional project templates](#)

**Authentication**  
No Authentication  
[Change](#)

**Advanced**  
☐ Enable Docker Support  
(Requires Docker Desktop)  

Linux

  
☒ ASP.NET Core hosted

Author: Microsoft  
Source: CLI v3.1.100-preview1-014459

Back

Create

# Solving the issue

- Go to project root, exec `dotnet --list-sdks` and copy the latest one
  - Execute `dotnet new globaljson --sdk-version {put here the copied .NET sdk version}`
  - Restart Visual Studio...
- 
- Or.. Use the preview for the best experience..





## Client-side (Blazor WebAssembly)

Preview now  
Release target: May 2020

## Server-side (Blazor Server)

Shipped  
Part of .NET Core 3.0

---

## Blazor + Electron (Desktop app with web rendering)

Uncommitted  
Sample available now

## Blazor + Native UI (Mobile/desktop with native OS UI)

Very experimental!  
Super-early, no sample yet

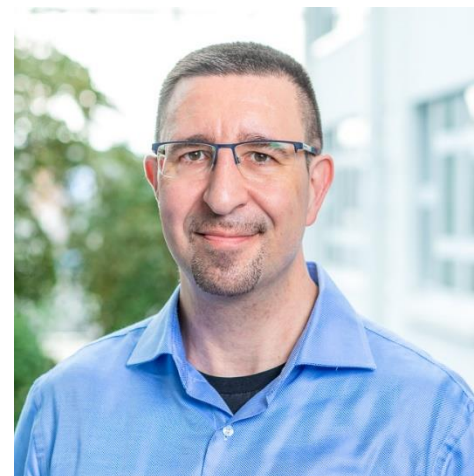
# Conclusion

## ■ 3 main takeaways:

1. Full .NET Standard 2.0
2. No JavaScript
3. Server-side Shipped & Client-side coming May 2020



# Questions?



**Jose Luis Latorre Millas**  
Lead Software Architect

[Jose.latorremillas@zuehlke.com](mailto:Jose.latorremillas@zuehlke.com) / [joslat@gmail.com](mailto:joslat@gmail.com)

Linkedin: [www.linkedin.com/in/joslat/](https://www.linkedin.com/in/joslat/)

Twitter: @joslat

Blog: [joslat.blog](http://joslat.blog)

Thanks .NET Zurich  
Developers UG! Hope  
you had fun 😊