



Using WebAssembly to run, extend, and secure your .NET application

Niels Tanis
Sr. Principal Security Researcher

GenetecTM
ElevateDev'24

Who am I?

- Niels Tanis
- Sr. Principal Security Researcher
 - Background .NET Development, Pentesting/ethical hacking, and software security consultancy
 - Research on static analysis for .NET apps
 - Enjoying Rust!
- Microsoft MVP – Developer Technologies

VERACODE

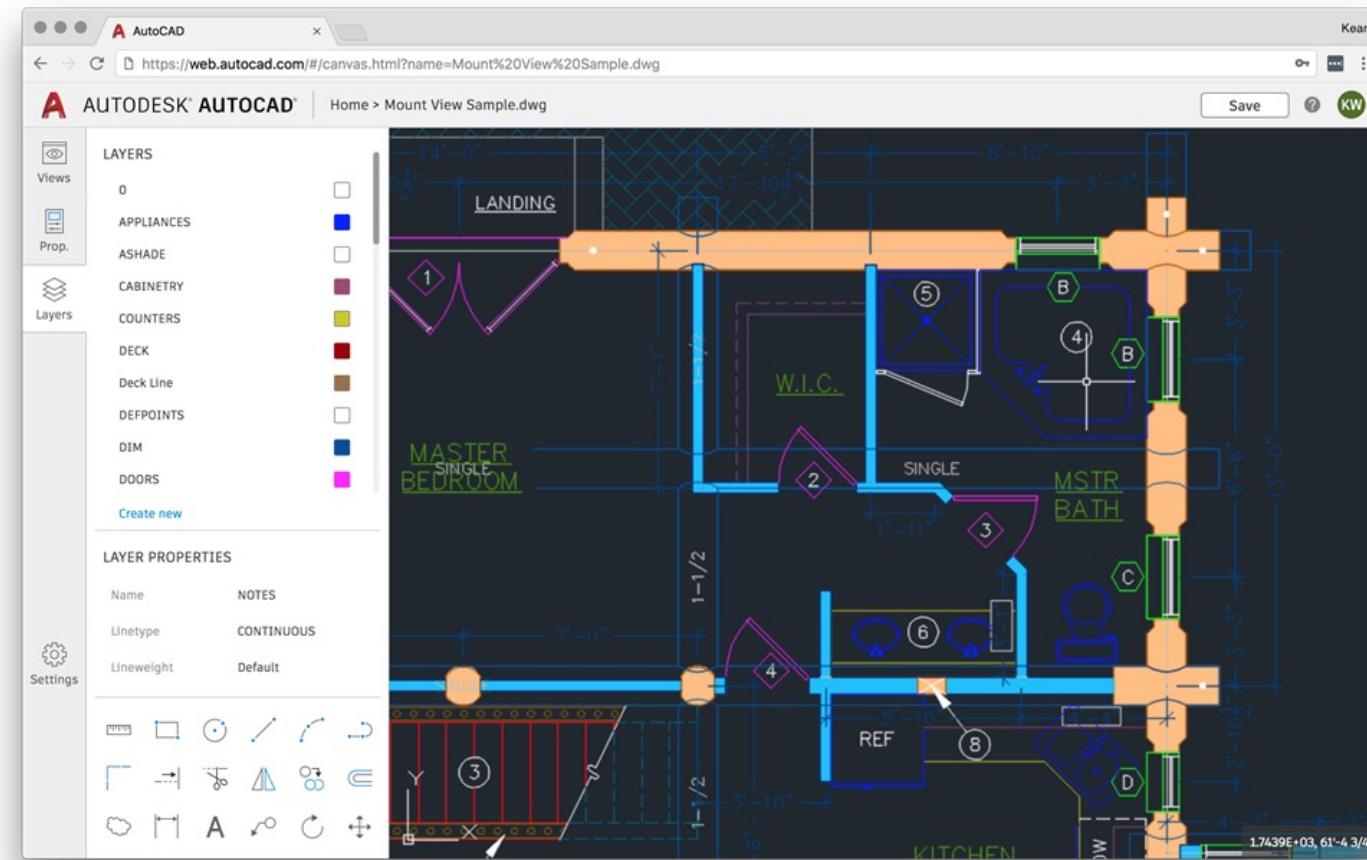


WebAssembly

The screenshot shows the official WebAssembly website (<https://webassembly.org>) displayed in a modern web browser. The page features a dark header with the title "WebAssembly" and a navigation bar with links to "Overview", "Getting Started", "Specs", "Feature Extensions", "Community", and "FAQ". Below the header is a large purple "WA" logo and the word "WEBASSEMBLY". A green banner at the top states "WebAssembly 1.0 has shipped in 4 major browser engines." followed by icons for Firefox, Chrome, Safari, and Edge, with a "Learn more" link. The main content area provides a brief introduction to WebAssembly, mentioning it is a binary instruction format for a stack-based virtual machine designed for portable compilation across programming languages. A yellow callout box at the bottom contains developer reference information, pointing to MDN's WebAssembly pages, the W3C Community Group, and the W3C Working Group.

Developer reference documentation for Wasm can be found on [MDN's WebAssembly pages](#). The open standards for WebAssembly are developed in a [W3C Community Group](#) (that includes representatives from all major browsers) as well as a [W3C Working Group](#).

WebAssembly - AutoCAD



WebAssembly - SDK's

A screenshot of a Medium article page. The title is "Introducing the Disney+ Application Development Kit (ADK)". It was published by Mike Hanley on Sep 8, 2021. The article has 415 upvotes and 4 comments. The content discusses the Disney+ Application Development Kit.

A screenshot of a Medium article page. The title is "How Prime Video updates its app for more than 8,000 device types". It was published by Alexandru Ene on January 27, 2022. The article discusses how Prime Video uses WebAssembly to support over 8,000 device types. It includes a section on "CLOUD AND SYSTEMS".

Agenda

- Introduction
- WebAssembly Design & Internals
- Running .NET on WebAssembly
- Extending .NET with WebAssembly
- Securing .NET with WebAssembly
- Conclusion
- Q&A

WebAssembly Design

- **Be fast, efficient, and portable**
 - Executed in near-native speed across different platforms
- **Be readable and debuggable**
 - In low-level bytecode but also human readable
- **Keep secure**
 - Run on sandboxed execution environment
- **Don't break the web**
 - Ensure backwards compatibility



WEBASSEMBLY

WebAssembly

- Binary instruction format for stack-based virtual machine similar to .NET CLR running MSIL or JVM running bytecode
- Designed as a portable compilation target



WEBASSEMBLY

WebAssembly

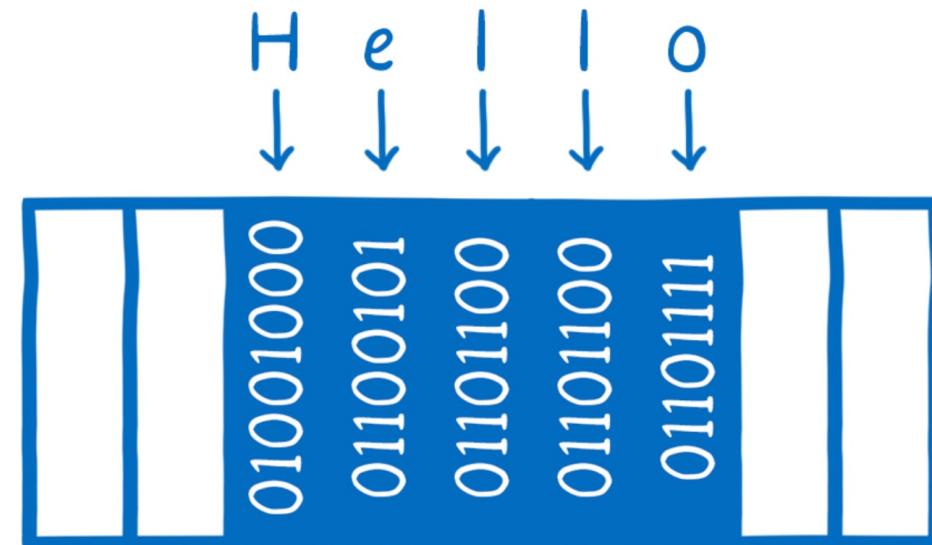
- The security model of WebAssembly:
 - Protect users from buggy or malicious modules
 - Provide developers with useful primitives and mitigations for developing safe applications



WEBASSEMBLY

WebAssembly Memory

- Isolated per WASM module
- A contiguous, mutable array of uninterpreted bytes



WebAssembly Control-Flow Integrity

```
int number = Convert.ToInt32(Console.ReadLine());
Console.WriteLine($"Number {number}");
if (number>5)
{
    Console.WriteLine("Number is larger than 5");
}
else
{
    Console.WriteLine("Number is smaller than 5");
}
Console.WriteLine("Done!");
```

WebAssembly Control-Flow Integrity

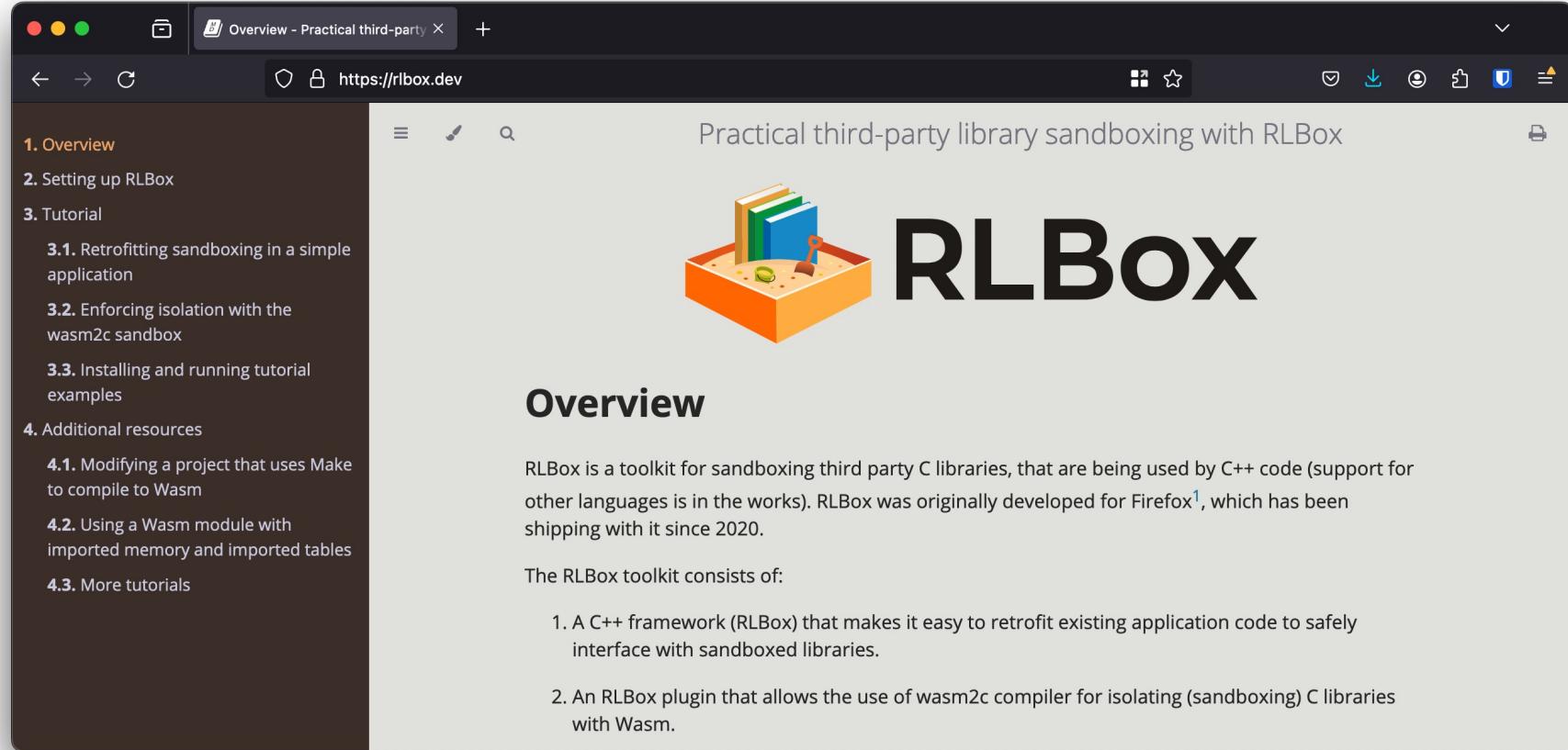
```
int number = Convert.ToInt32(Console.ReadLine());
Console.WriteLine($"Number {number}");
if (number>5)
```

```
Console.WriteLine("Number is larger than 5");
```

```
Console.WriteLine("Number is smaller than 5");
```

```
Console.WriteLine("Done!");
```

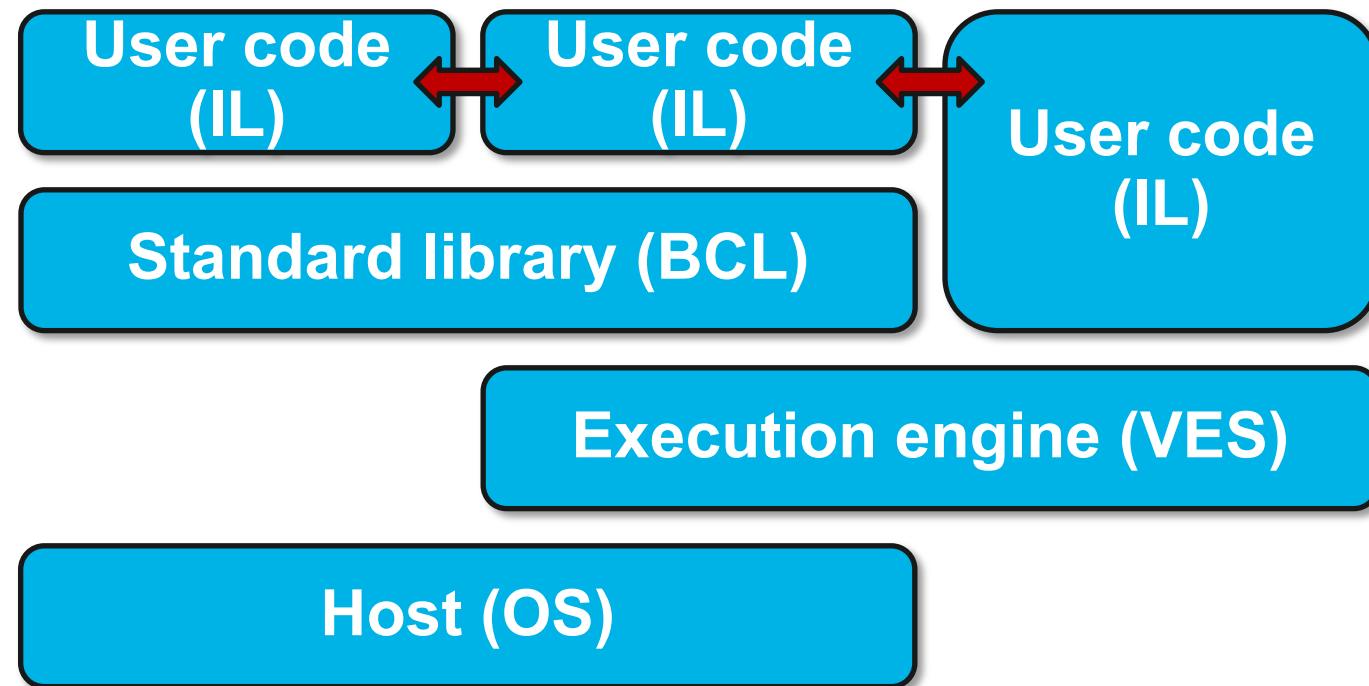
FireFox RLBox



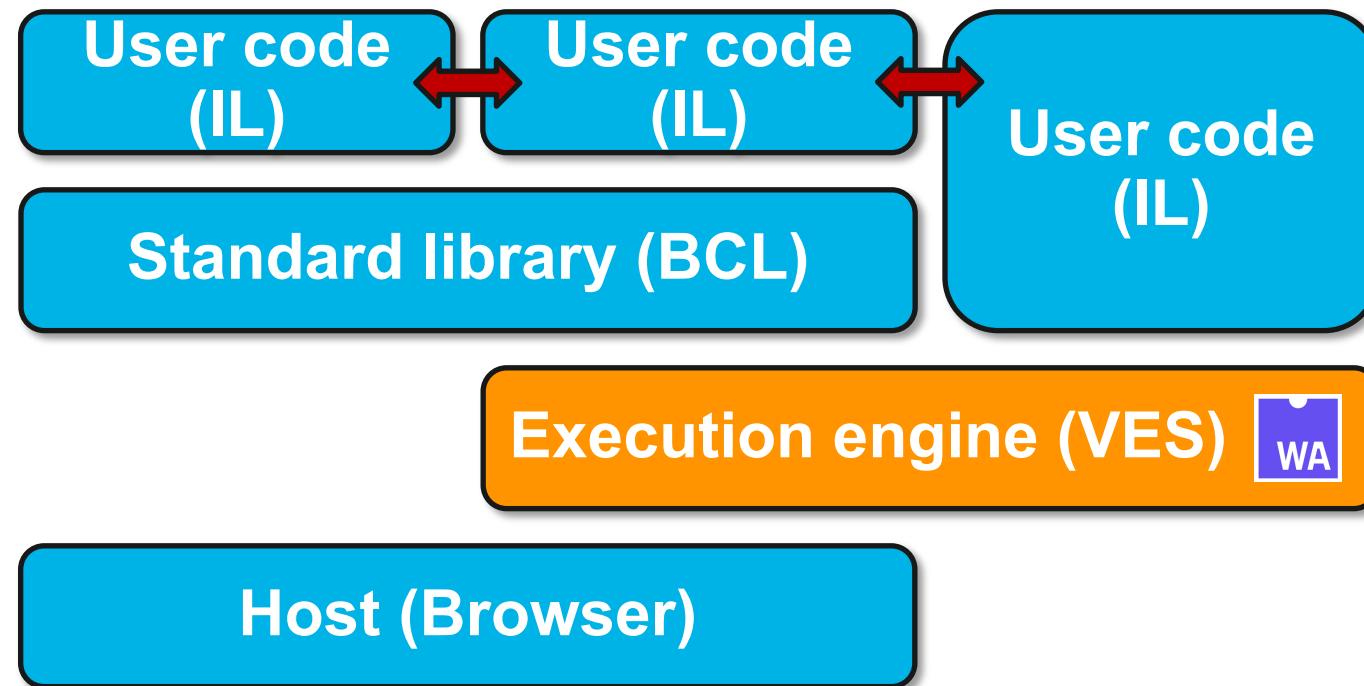
The screenshot shows a Firefox browser window with the URL <https://rlbox.dev>. The page title is "Overview - Practical third-party". The main content area features a logo of three books in a sandbox with a shovel, followed by the text "Practical third-party library sandboxing with RLBox" and the large "RLBox" logo. Below this, there is a section titled "Overview" with a brief description of what RLBox is and its history. To the left, a sidebar contains a navigation menu:

- 1. Overview
- 2. Setting up RLBox
- 3. Tutorial
 - 3.1. Retrofitting sandboxing in a simple application
 - 3.2. Enforcing isolation with the wasm2c sandbox
 - 3.3. Installing and running tutorial examples
- 4. Additional resources
 - 4.1. Modifying a project that uses Make to compile to Wasm
 - 4.2. Using a Wasm module with imported memory and imported tables
 - 4.3. More tutorials

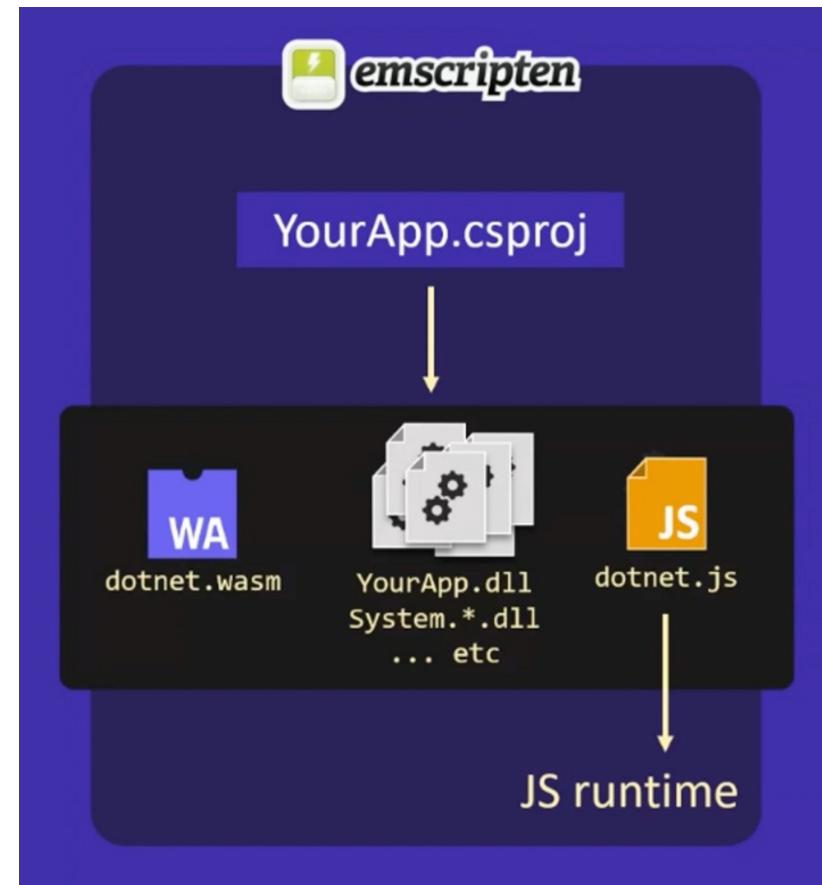
Running .NET on WebAssembly



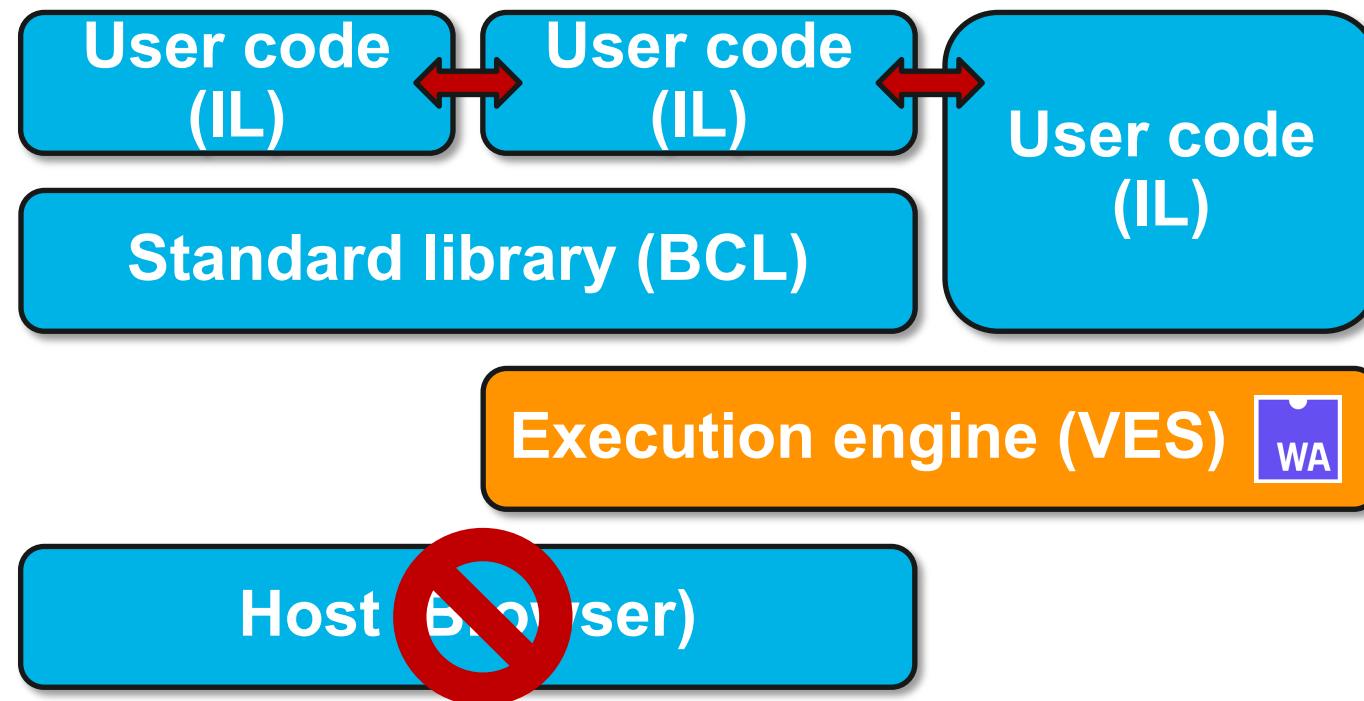
Running .NET on WebAssembly



Blazor WebAssembly



Running .NET on WebAssembly



WebAssembly System Interface WASI

- Introduced in March 2019 by Bytecode Alliance
- WasmTime implementation as reference
- POSIX inspired, engine-independent, non-Web system-oriented API for WebAssembly

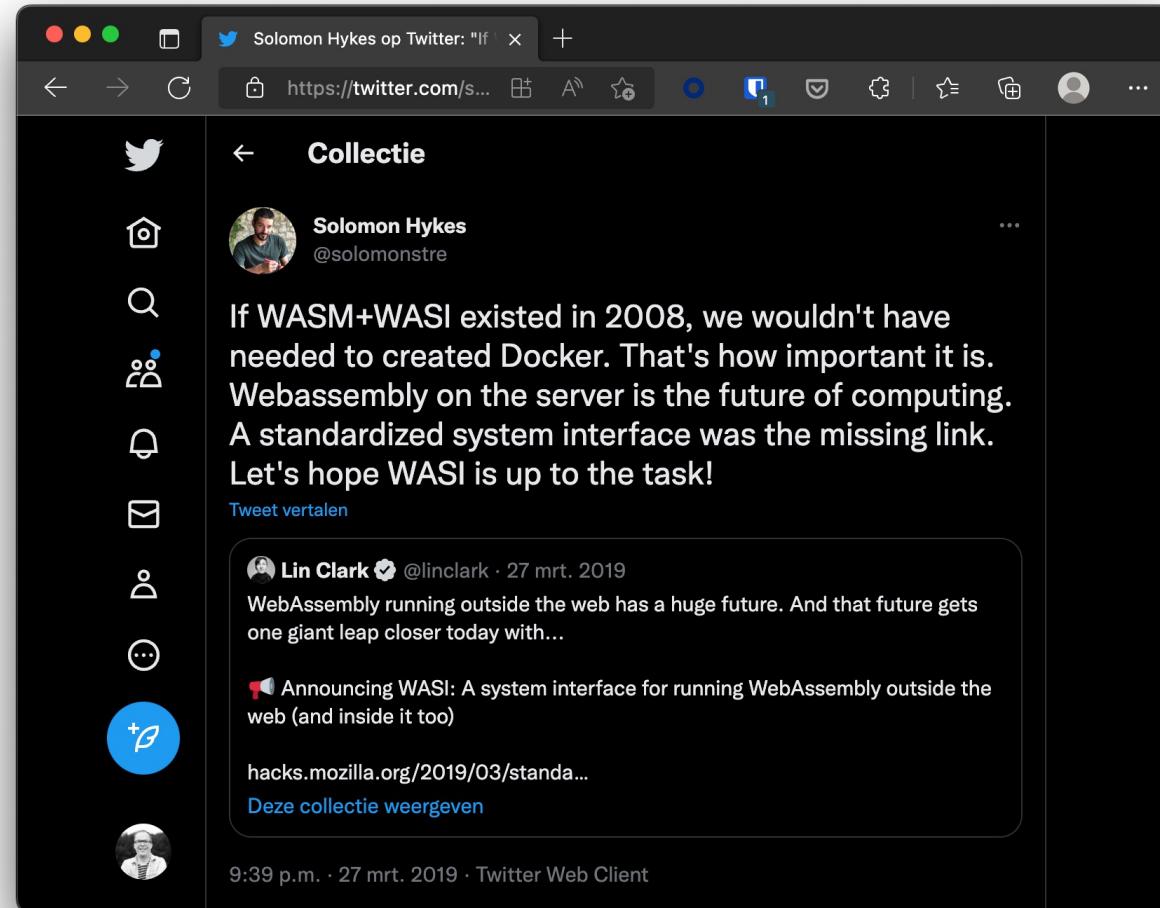


WebAssembly System Interface WASI

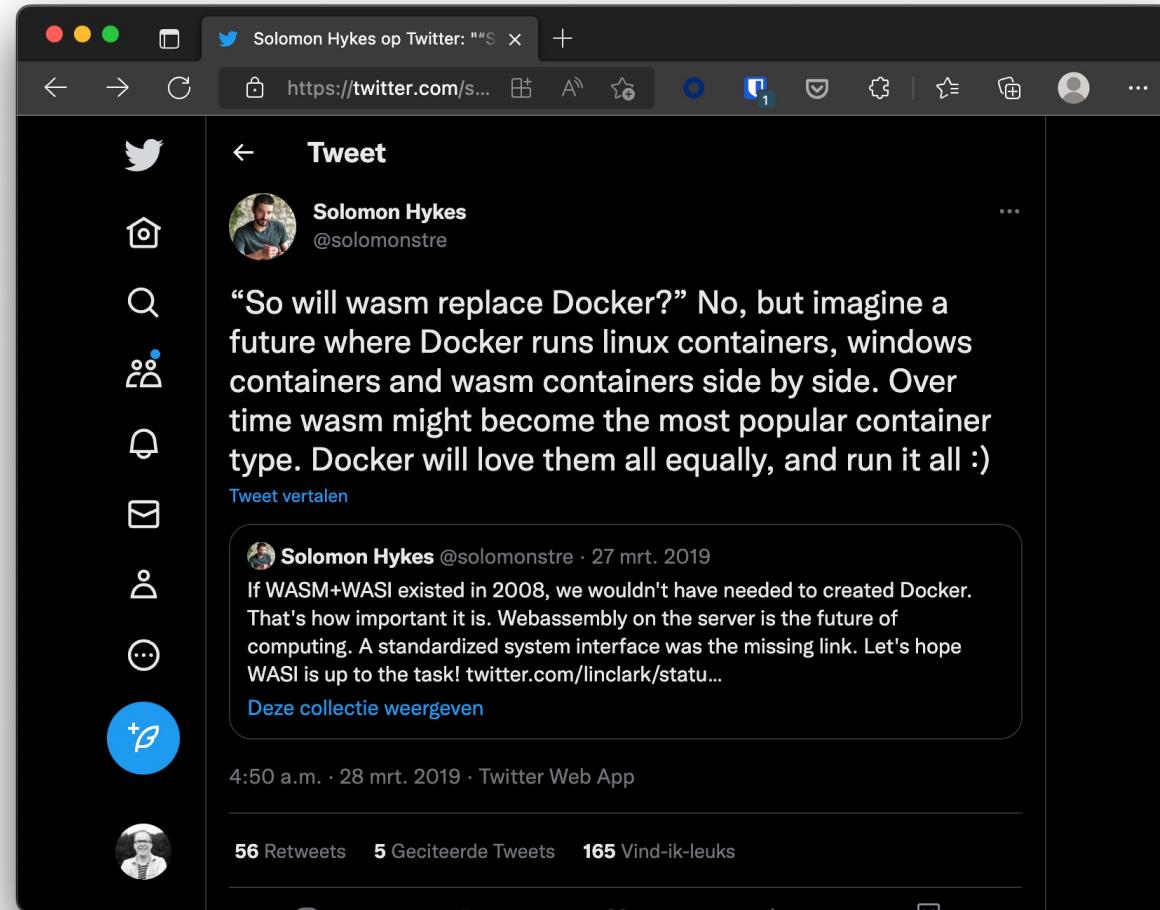
- Strong sandbox with Capability Based Security
- Preview1, supports e.g. FileSystem actions
- Future support for sockets and other system resources.
- Anyone recall .NET Standard? ☺



Docker vs WASM & WASI



Docker vs WASM & WASI



Docker & WASM

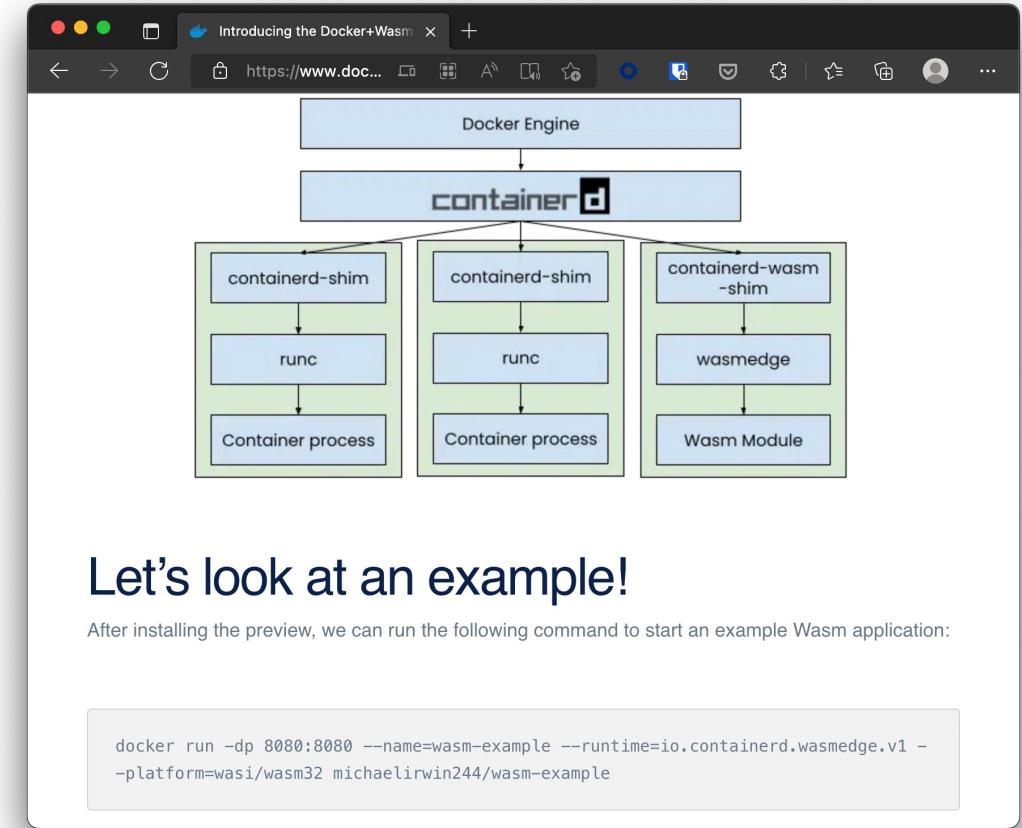
The screenshot shows a web browser window with the title "Introducing the Docker+Wasm Technical Preview". The page features a purple header bar with the text "Wasm is a fast, light alternative to Linux containers — try it out today in the Docker+Wasm Technical Preview". Below this is the Docker+Wasm logo. The main content area has a dark blue background with the heading "Introducing the Docker+Wasm Technical Preview" in large white font. Below the heading is a photo of Michael Irwin, a bio section, and a timestamp "Oct 24 2022". At the bottom, there is a paragraph about the availability of the Technical Preview.

Wasm is a fast, light alternative to Linux containers — try it out today in the [Docker+Wasm Technical Preview](#)

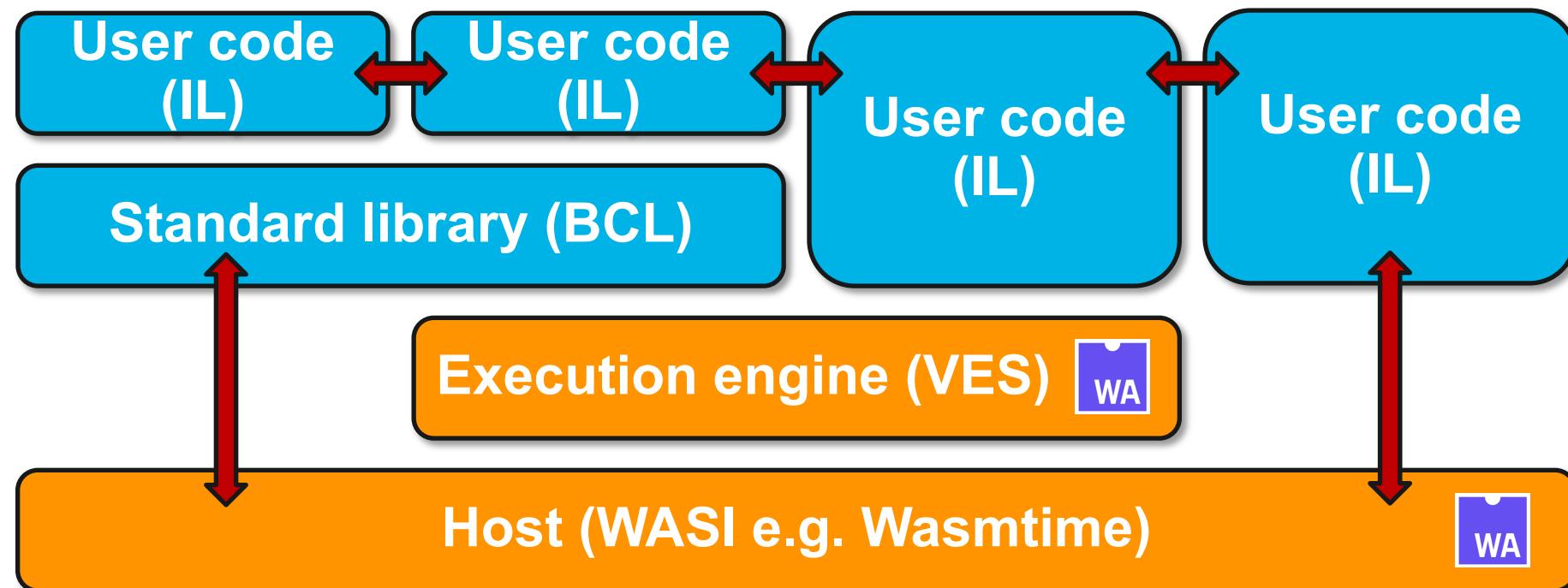
 MICHAEL IRWIN

Oct 24 2022

The [Technical Preview of Docker+Wasm](#) is now available! Wasm has been producing a lot of buzz recently, and this feature will make it easier for you to quickly build applications targeting Wasm runtimes.



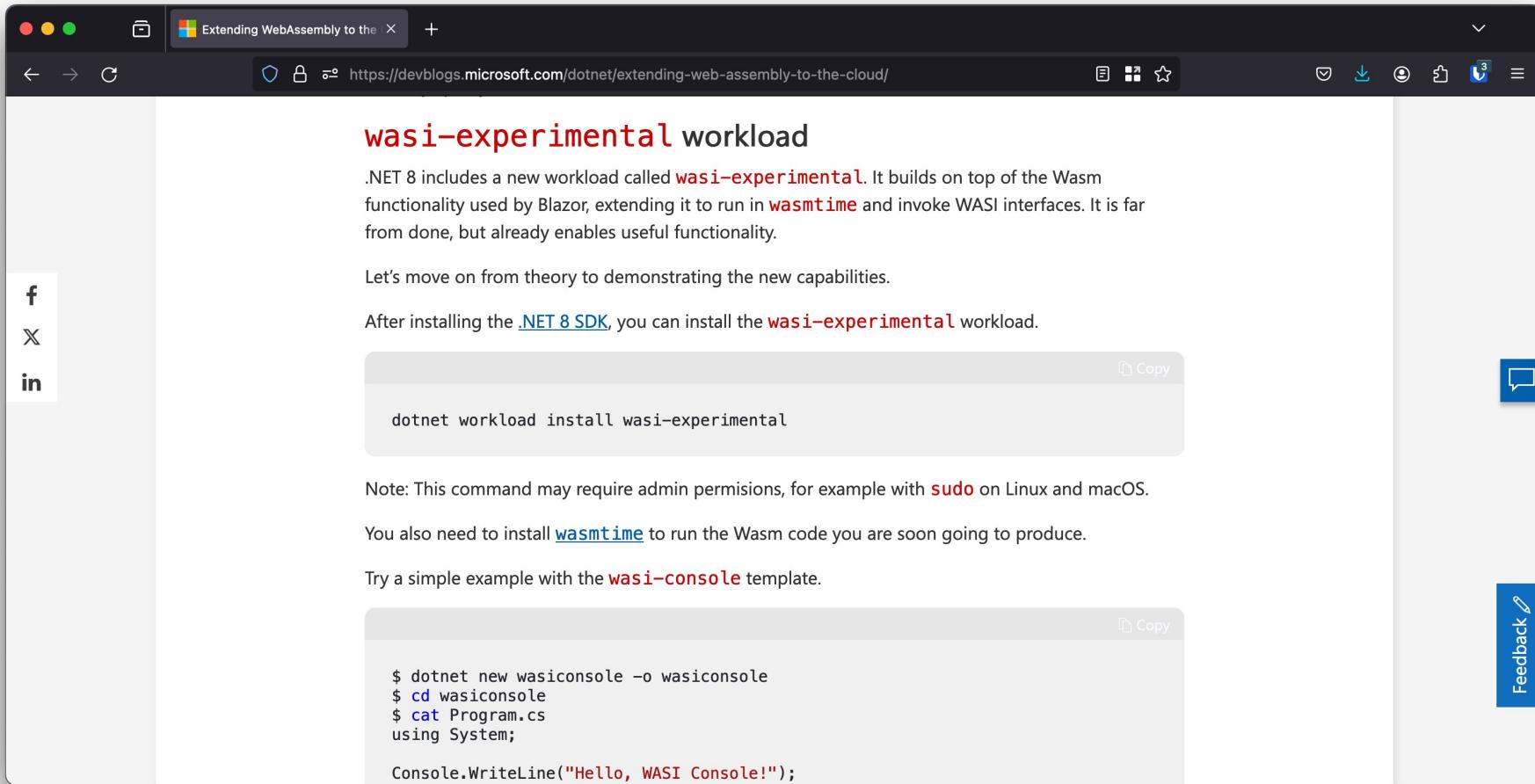
WebAssembly System Interface WASI



Experimental WASI SDK for .NET



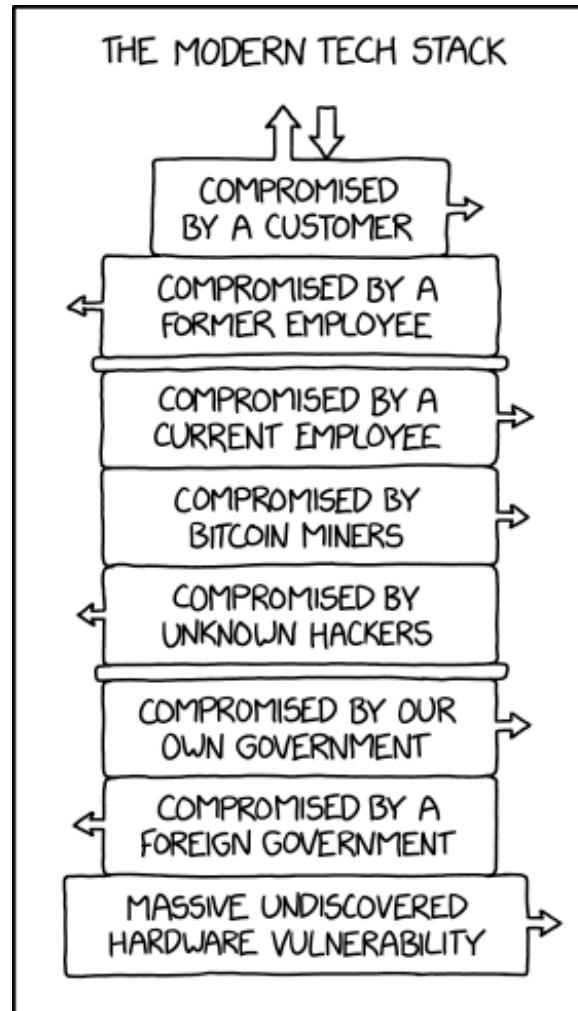
.NET 8 WASI-Experimental



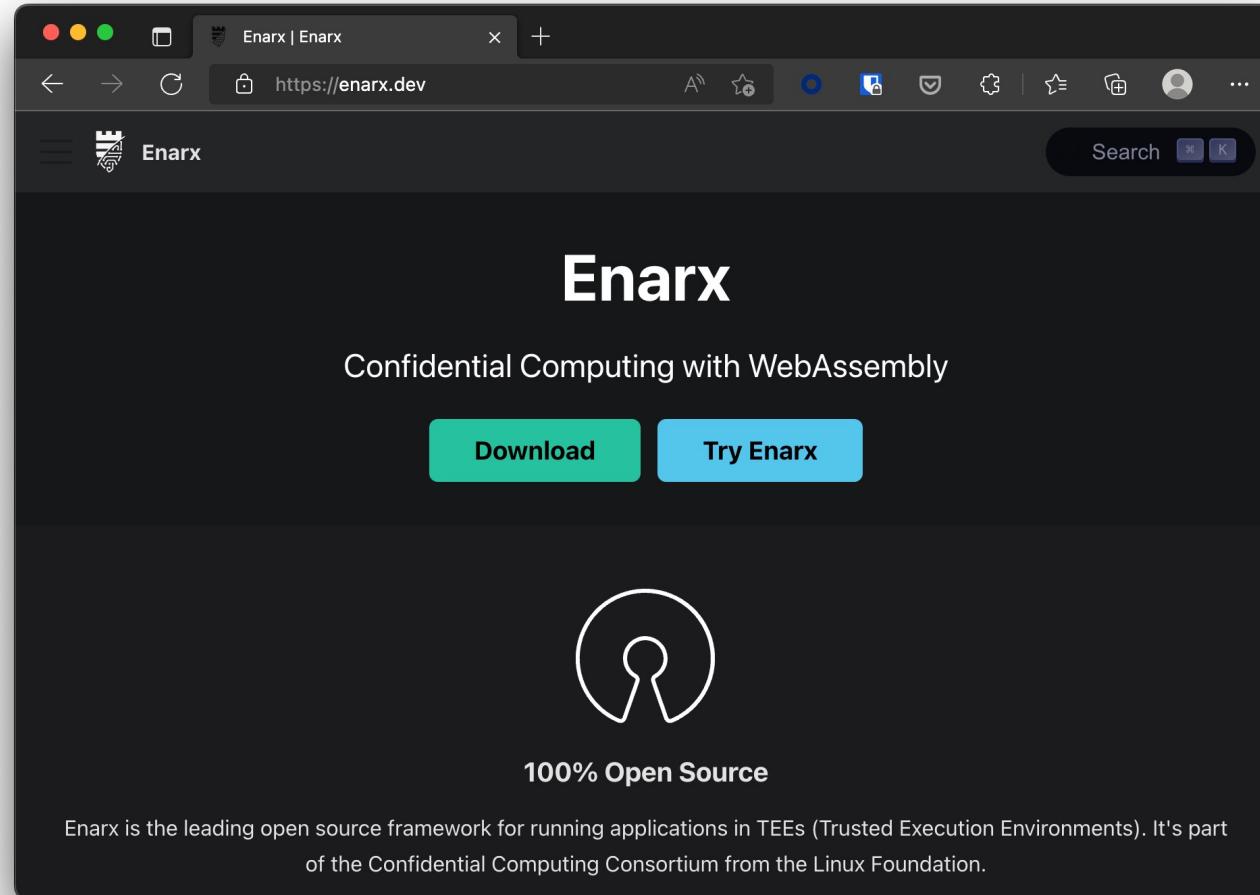
Extending .NET with WASM

- WasmTime.NET NuGet package
- Can run WASM inside of any .NET application
- Extend with Rust based WASM module
- Limit capabilities
- Demo time!

Trusted Computing - XKCD 2166

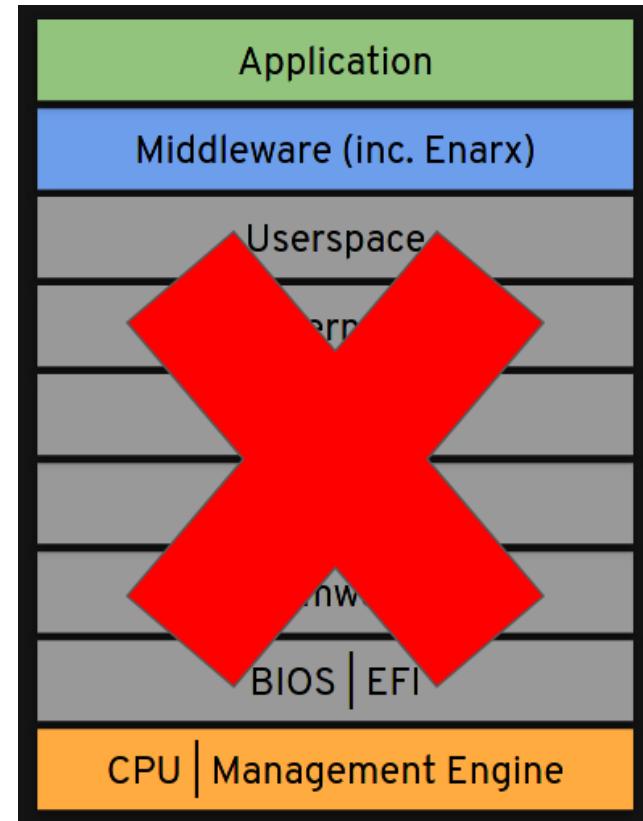


Enarx



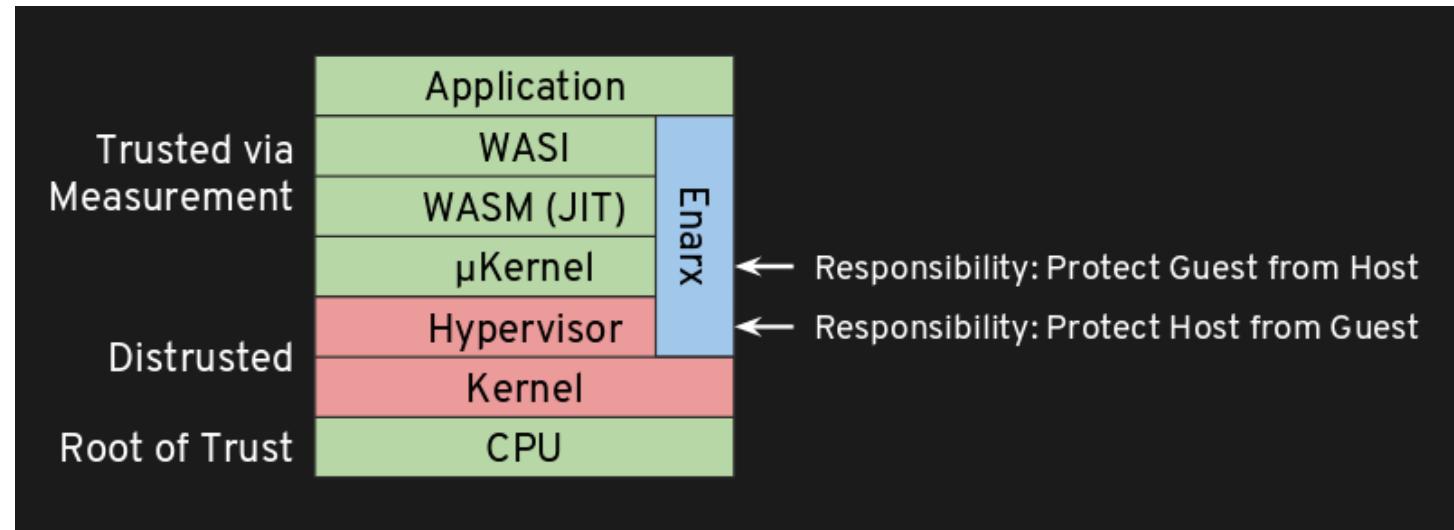
Enarx Threat Model

- Don't trust the host
- Don't trust the host owner
- Don't trust the host operator
- Hardware cryptographically verified
- Software audited and cryptographically verified

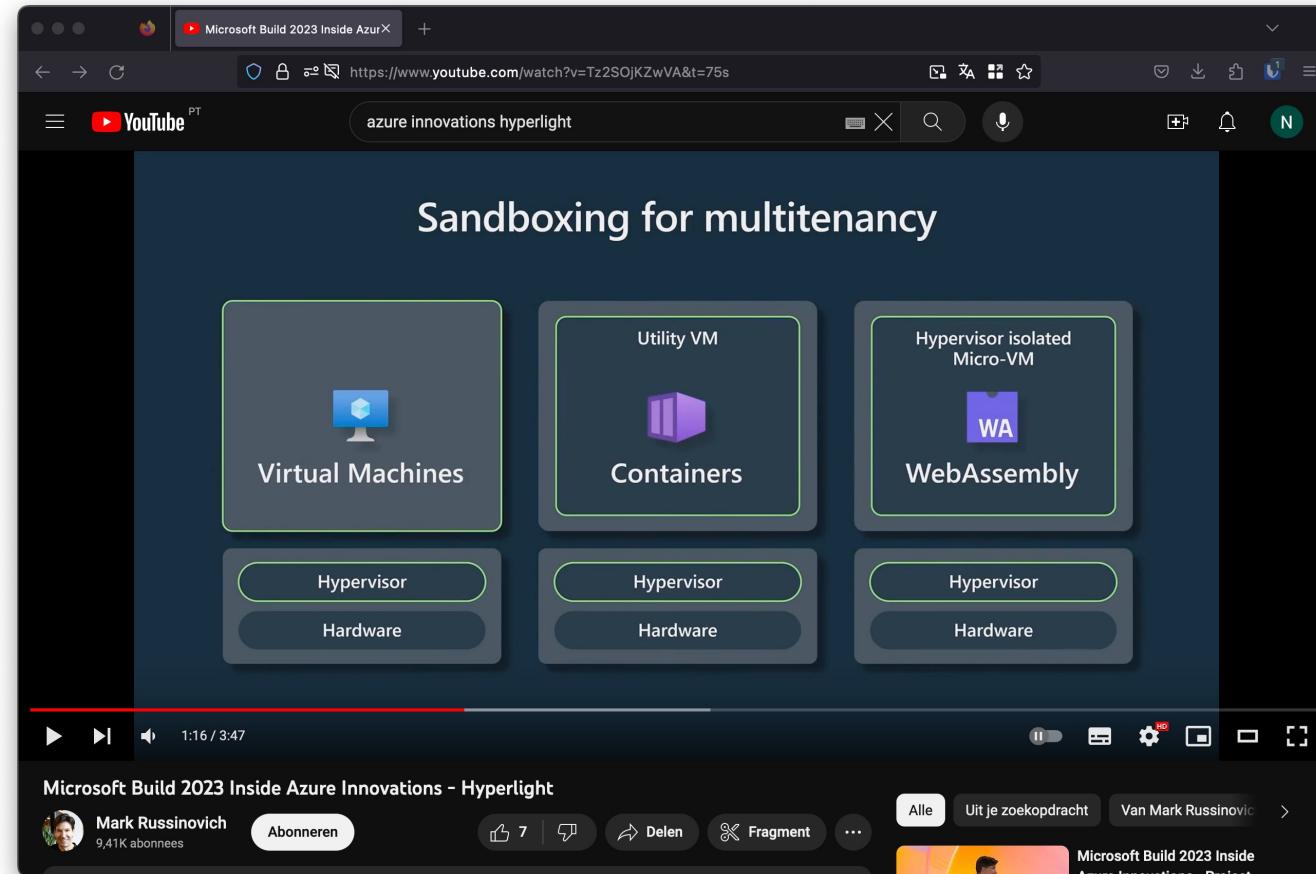


Enarx

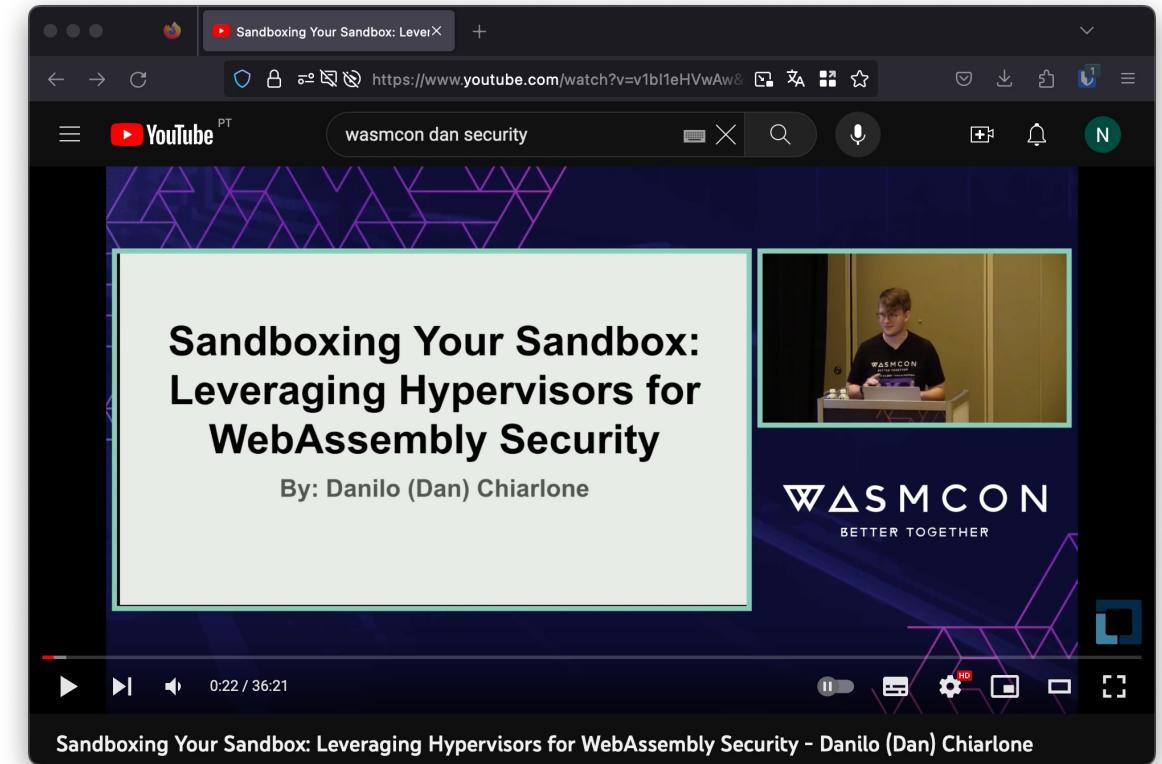
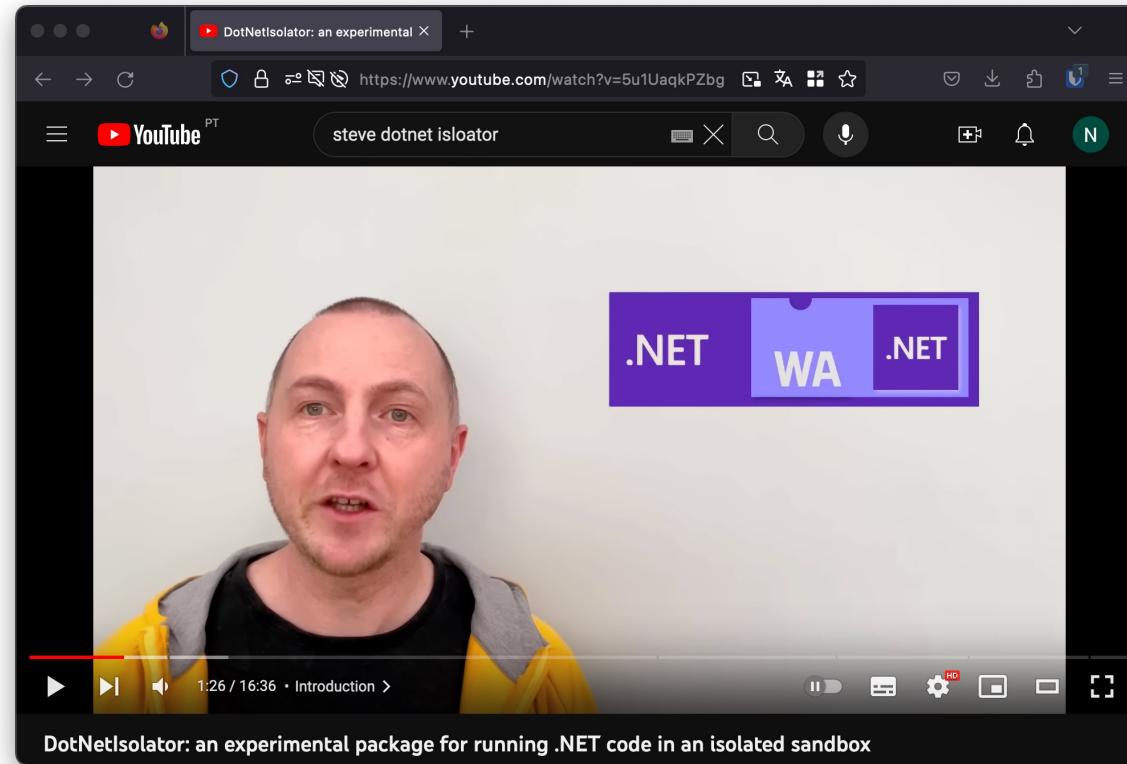
- Leverages Trusted Execution Environment (TEE) direct on processor
 - AMD's SEV, Intel's SGX and IBM's PEF
- Attestation of hardware and Enarx runtime



Project Hyperlight

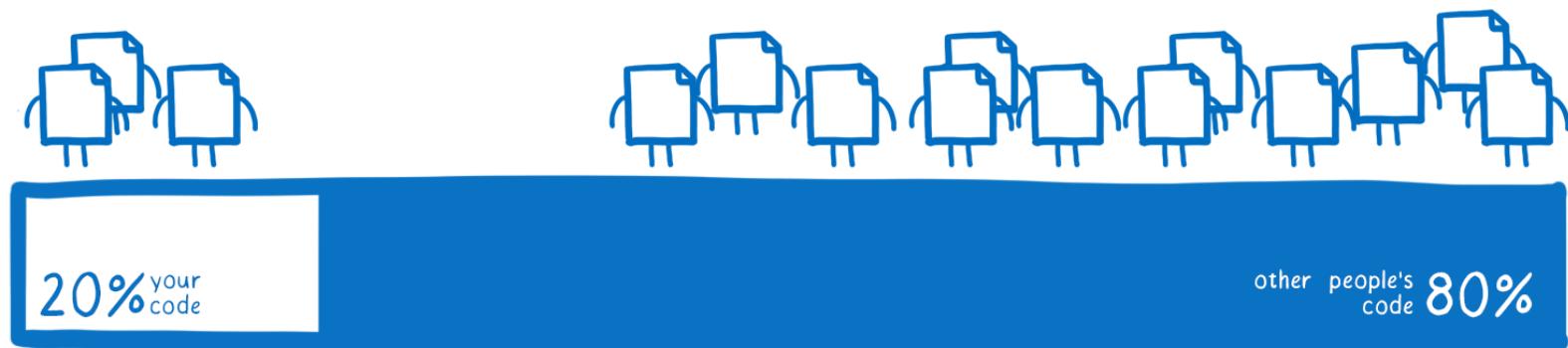


DotNetIsolator & Project Hyperlight

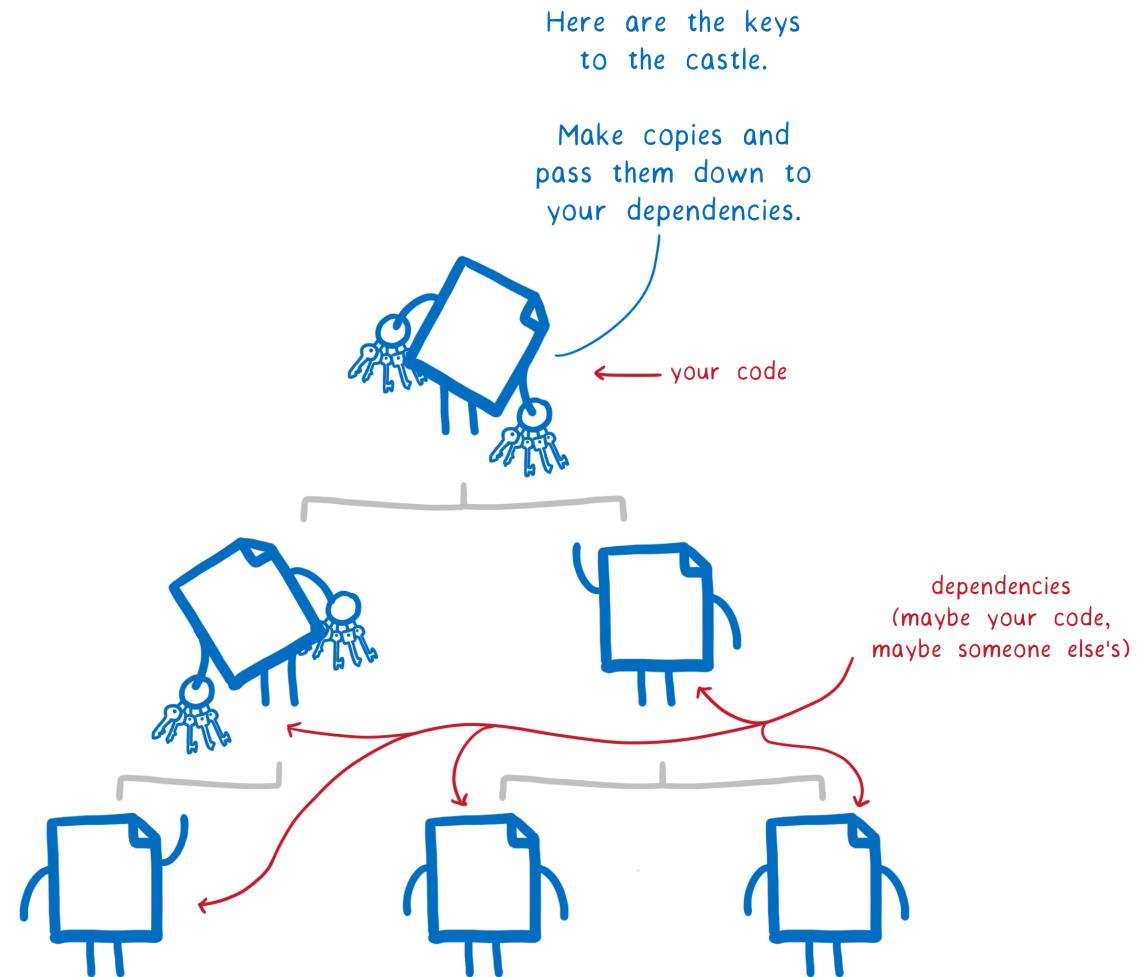


WASM - What's next?

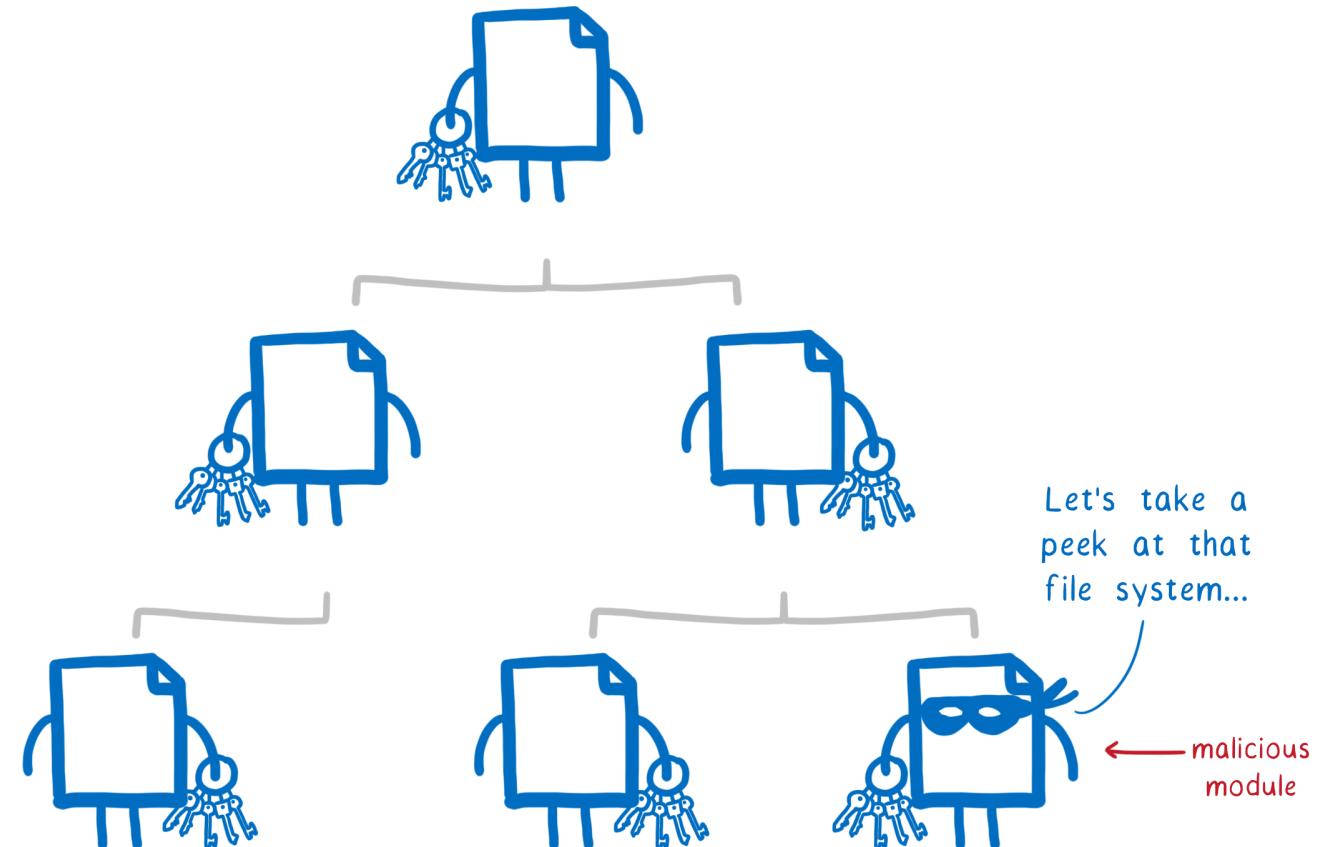
composition of an
average code base



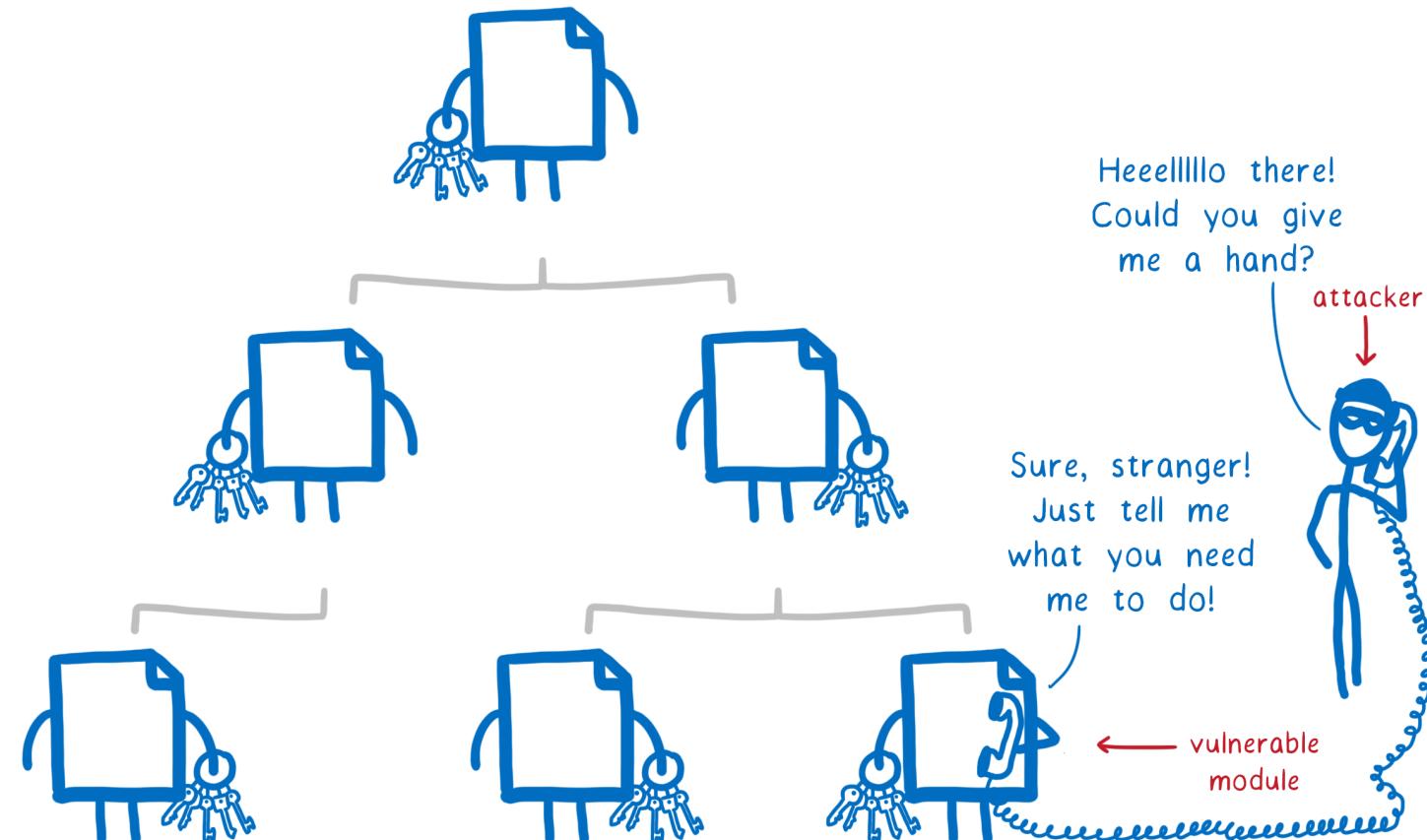
Dependencies



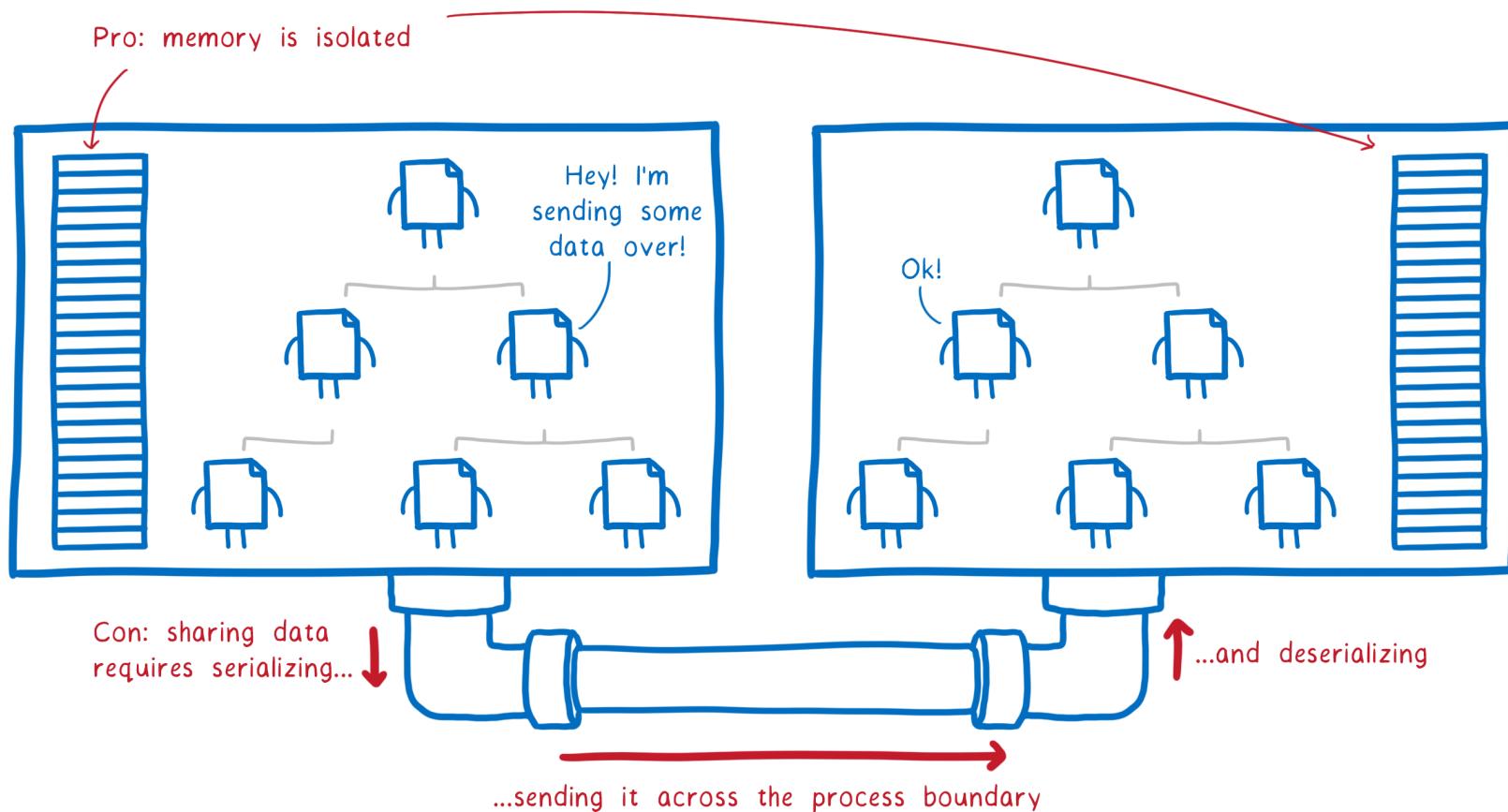
Malicious module



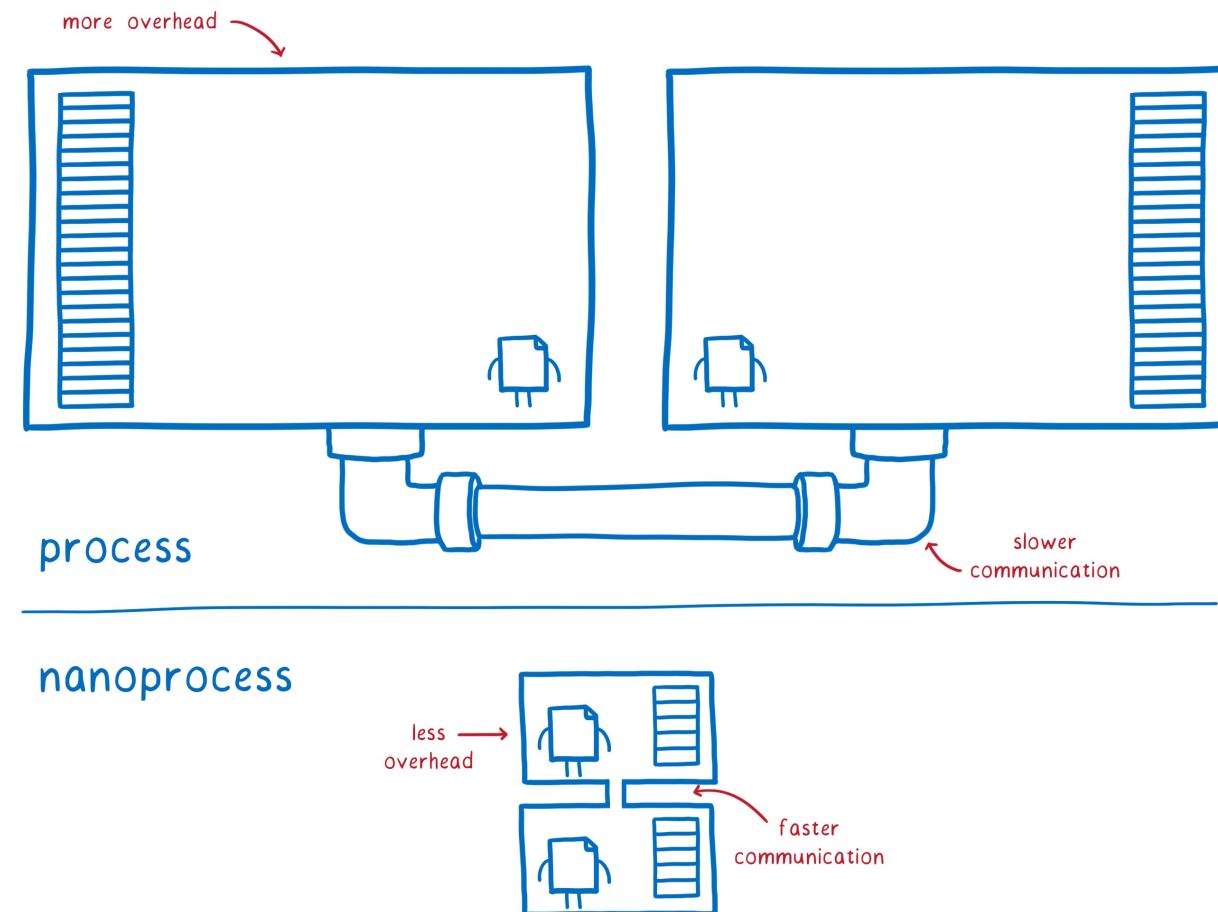
Vulnerable module



Process Isolation

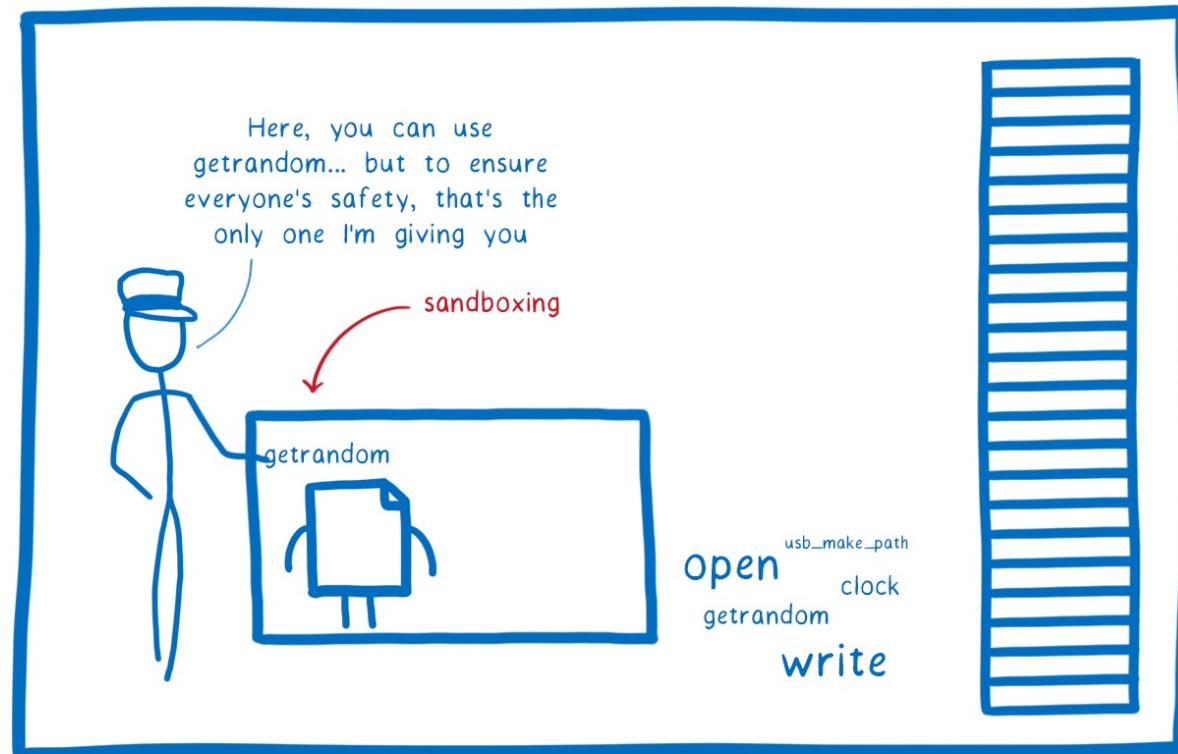


WebAssembly Nano-Process



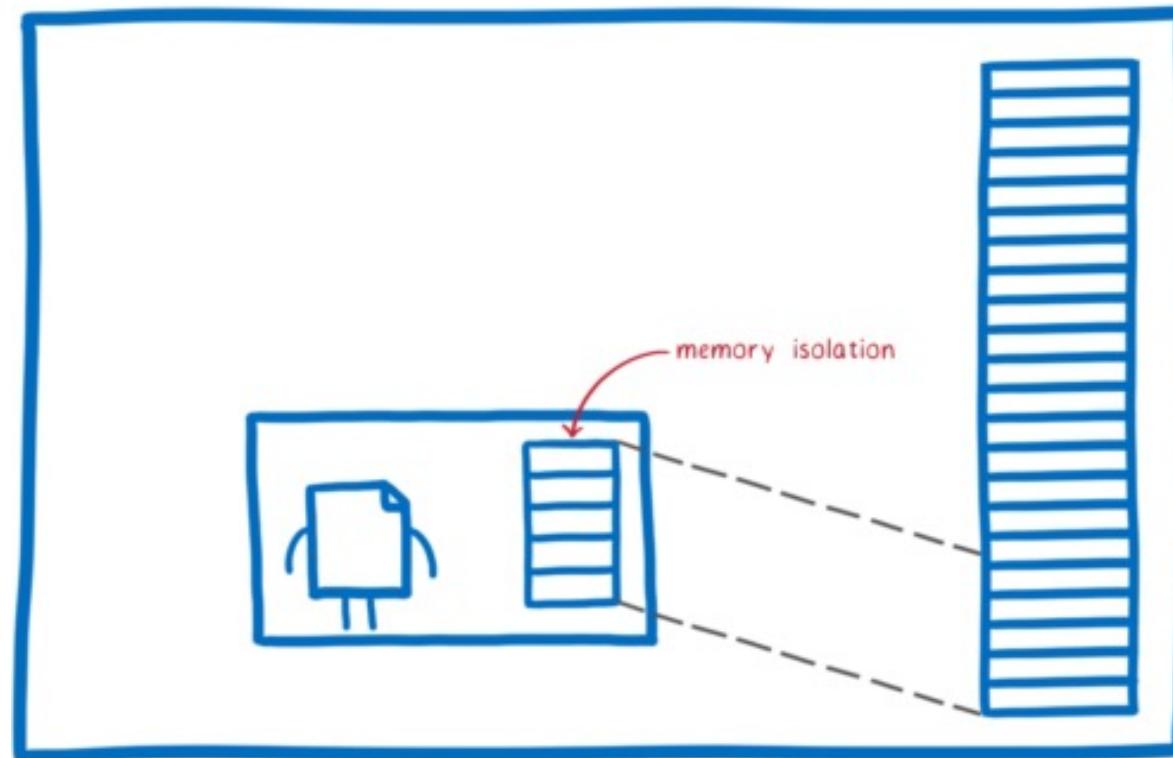
WebAssembly Nano-Process

1. Sandboxing



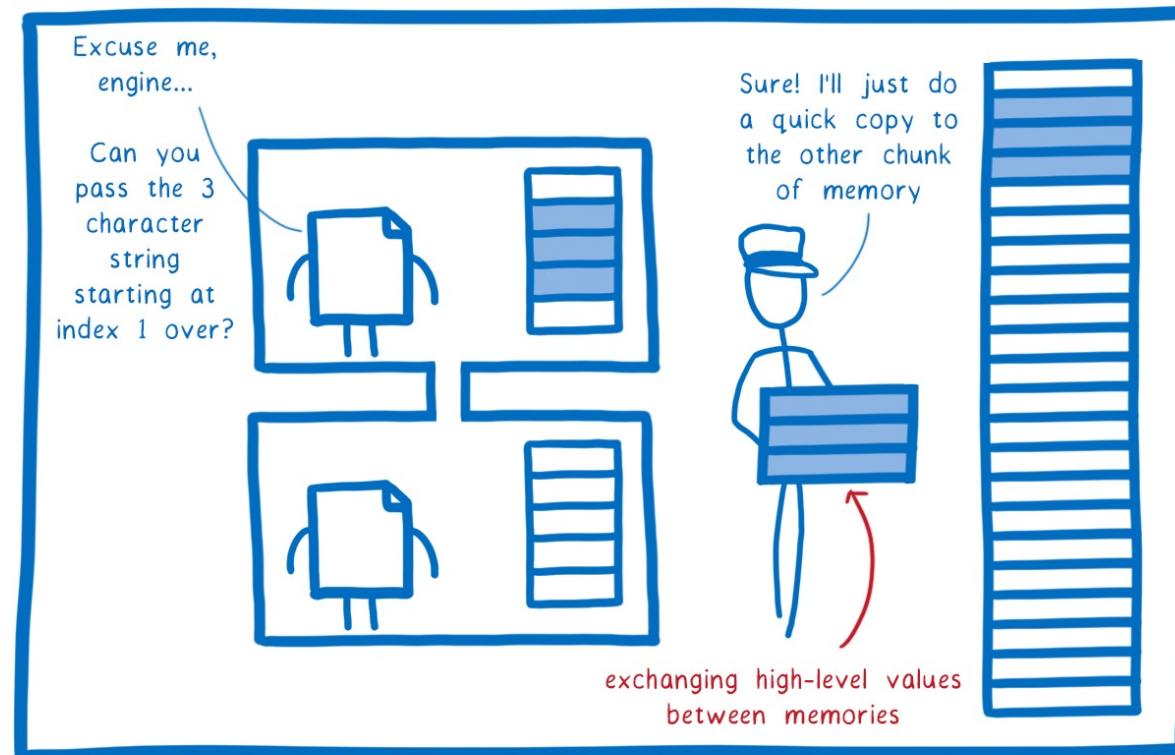
WebAssembly Nano-Process

2. Memory model



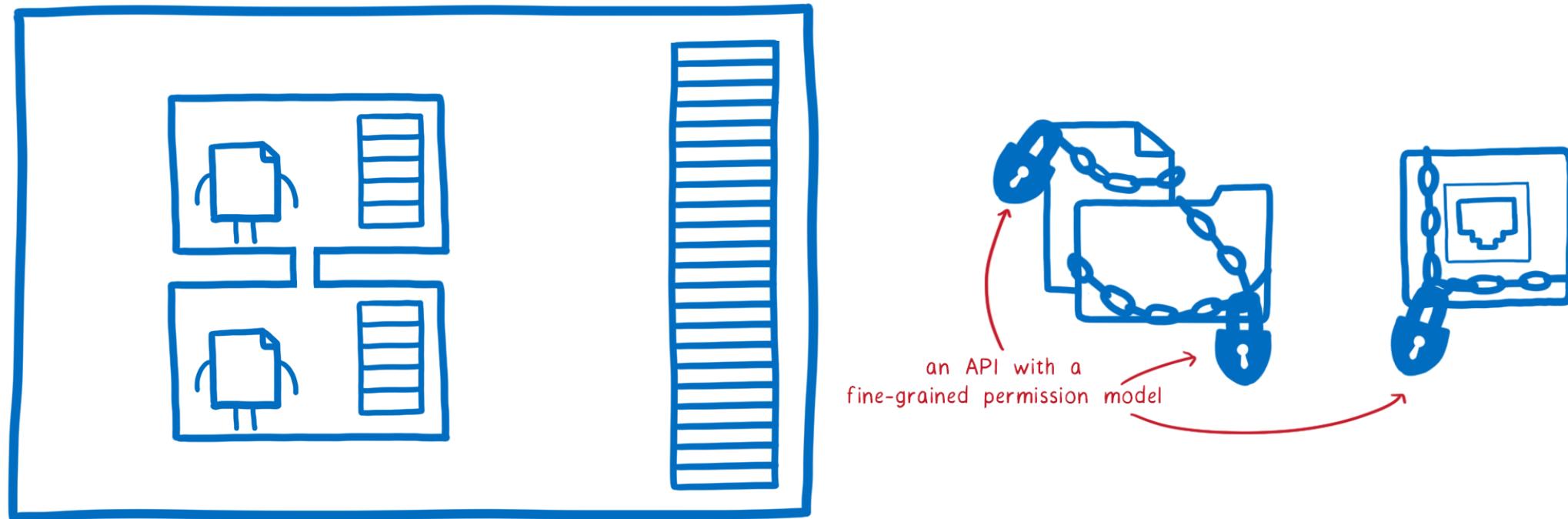
WebAssembly Nano-Process

3. Interface Types



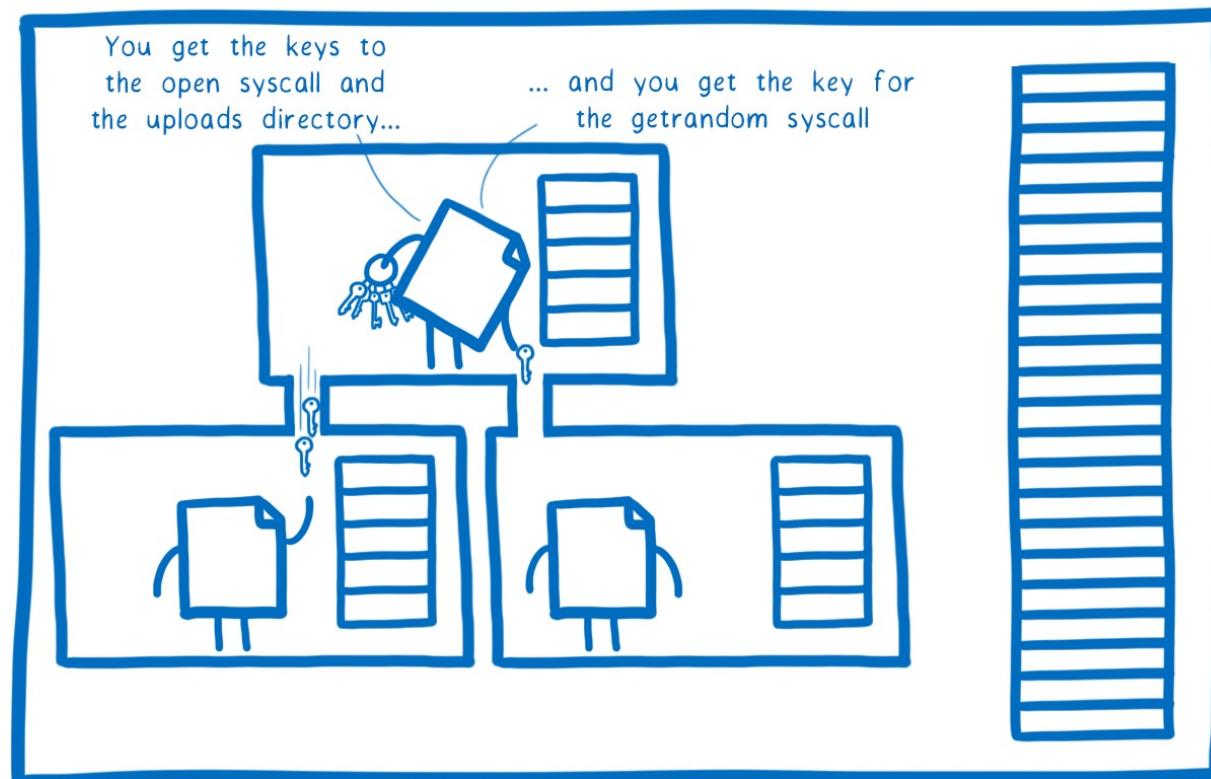
WebAssembly Nano-Process

4. WebAssembly System Interface

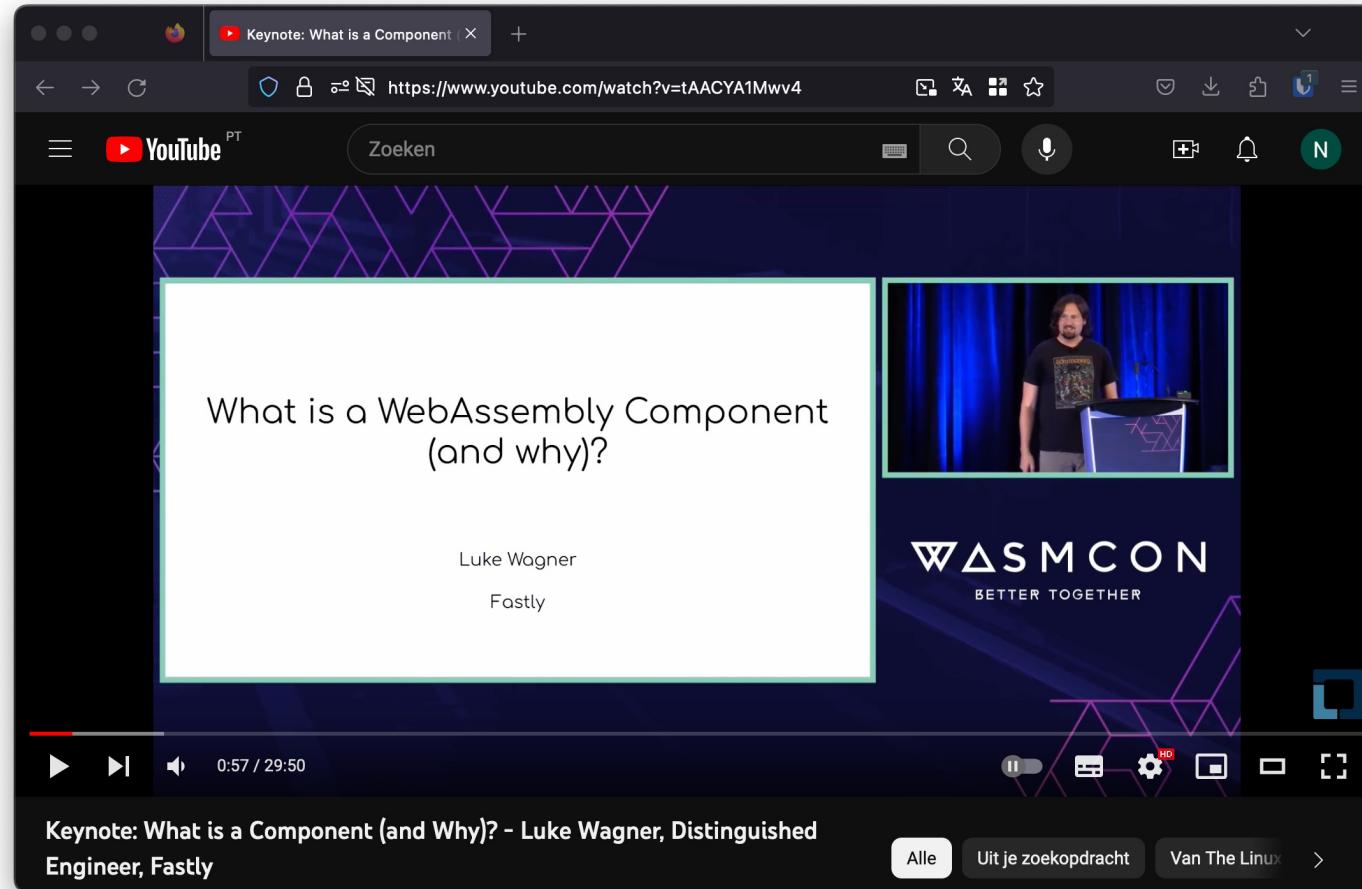


WebAssembly Nano-Process

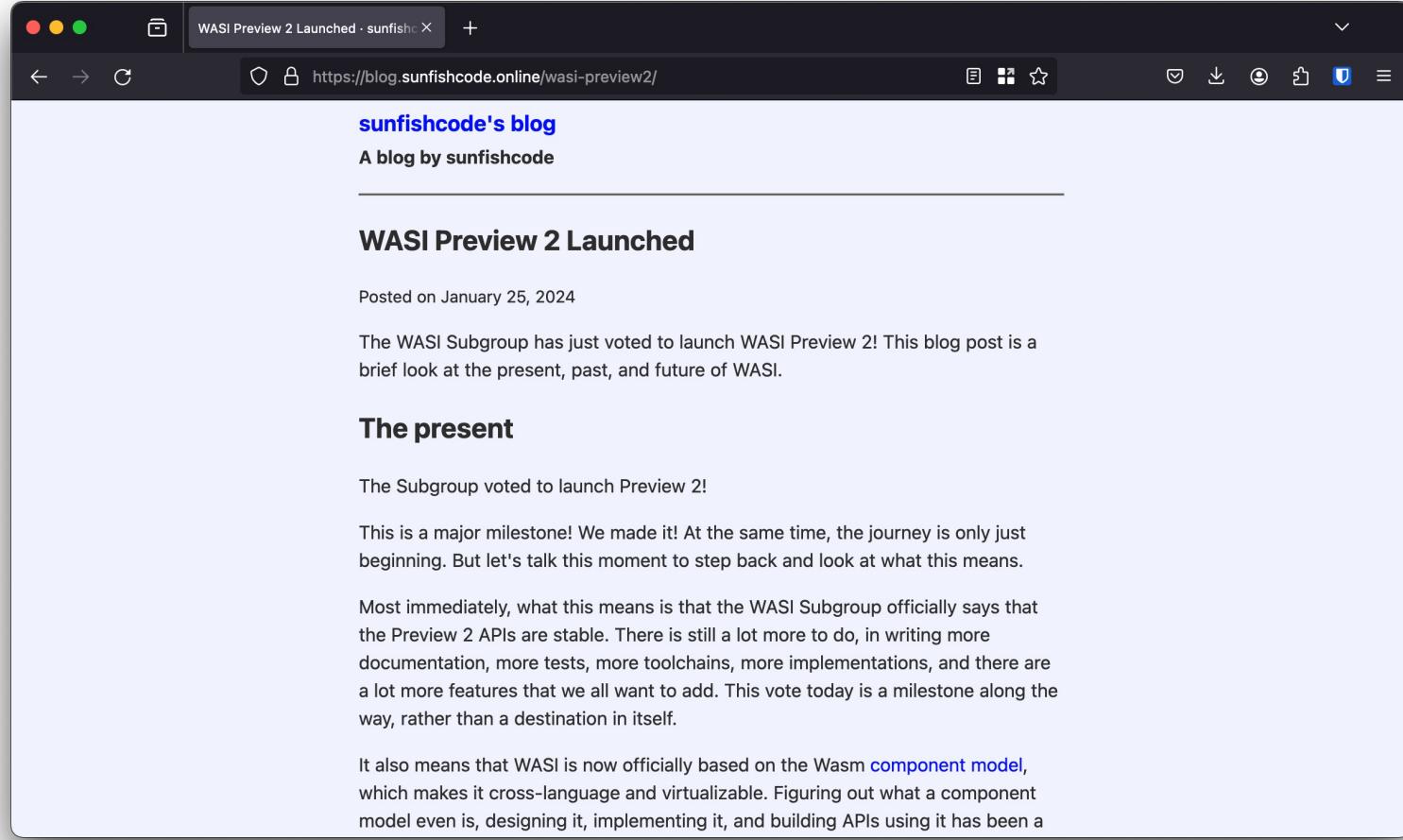
5. The missing link



WebAssembly Component Model



WASI Preview 2



The screenshot shows a web browser window titled "WASI Preview 2 Launched - sunfishcode". The URL in the address bar is <https://blog.sunfishcode.online/wasi-preview2/>. The page content is from "sunfishcode's blog" and is titled "A blog by sunfishcode". The main article is titled "WASI Preview 2 Launched" and was posted on January 25, 2024. The text discusses the launch of WASI Preview 2 and its significance as a major milestone. It highlights the stability of the Preview 2 APIs and the ongoing work required for documentation, testing, and implementation. The text also mentions the Wasm component model and its cross-language and virtualizable nature.

WASI Preview 2 Launched

Posted on January 25, 2024

The WASI Subgroup has just voted to launch WASI Preview 2! This blog post is a brief look at the present, past, and future of WASI.

The present

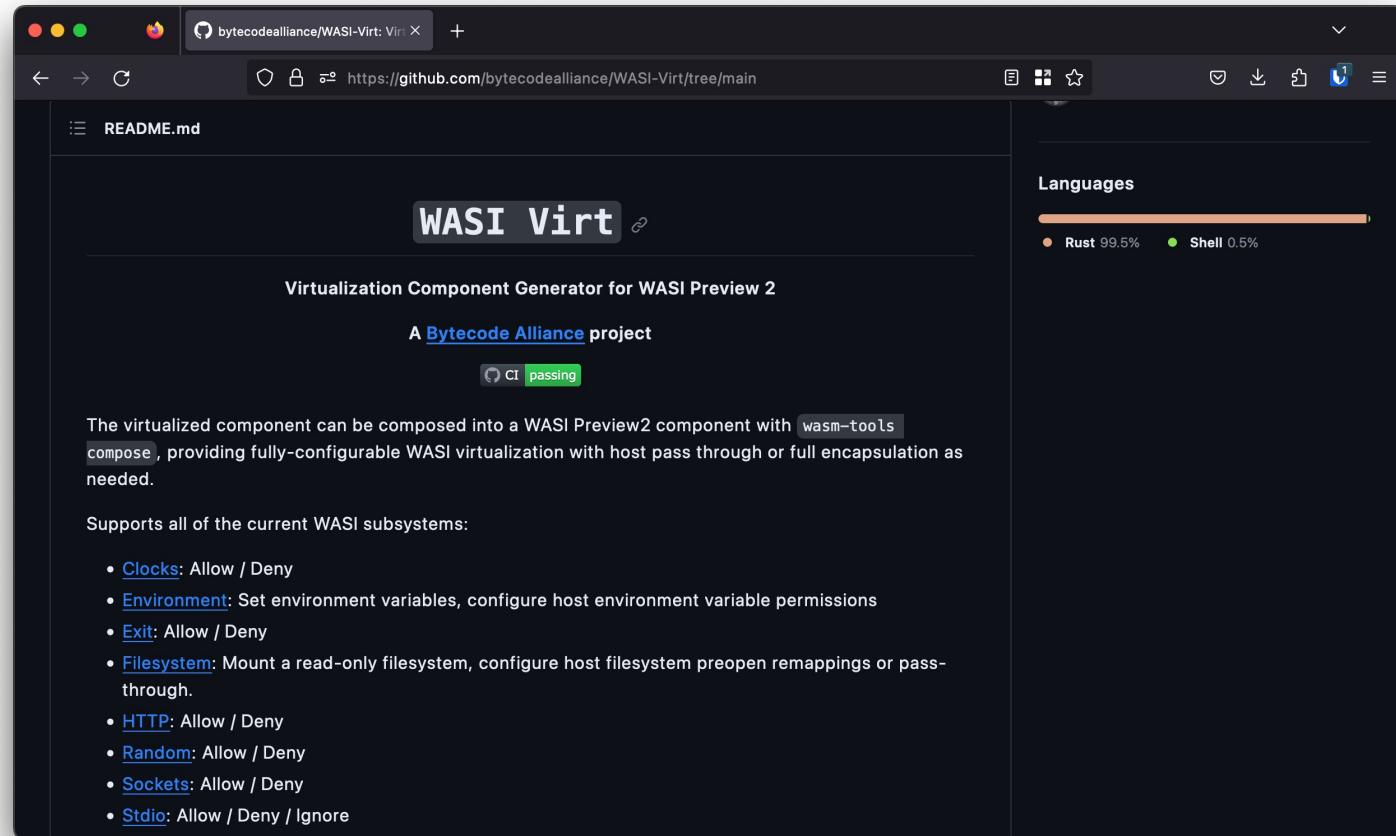
The Subgroup voted to launch Preview 2!

This is a major milestone! We made it! At the same time, the journey is only just beginning. But let's talk this moment to step back and look at what this means.

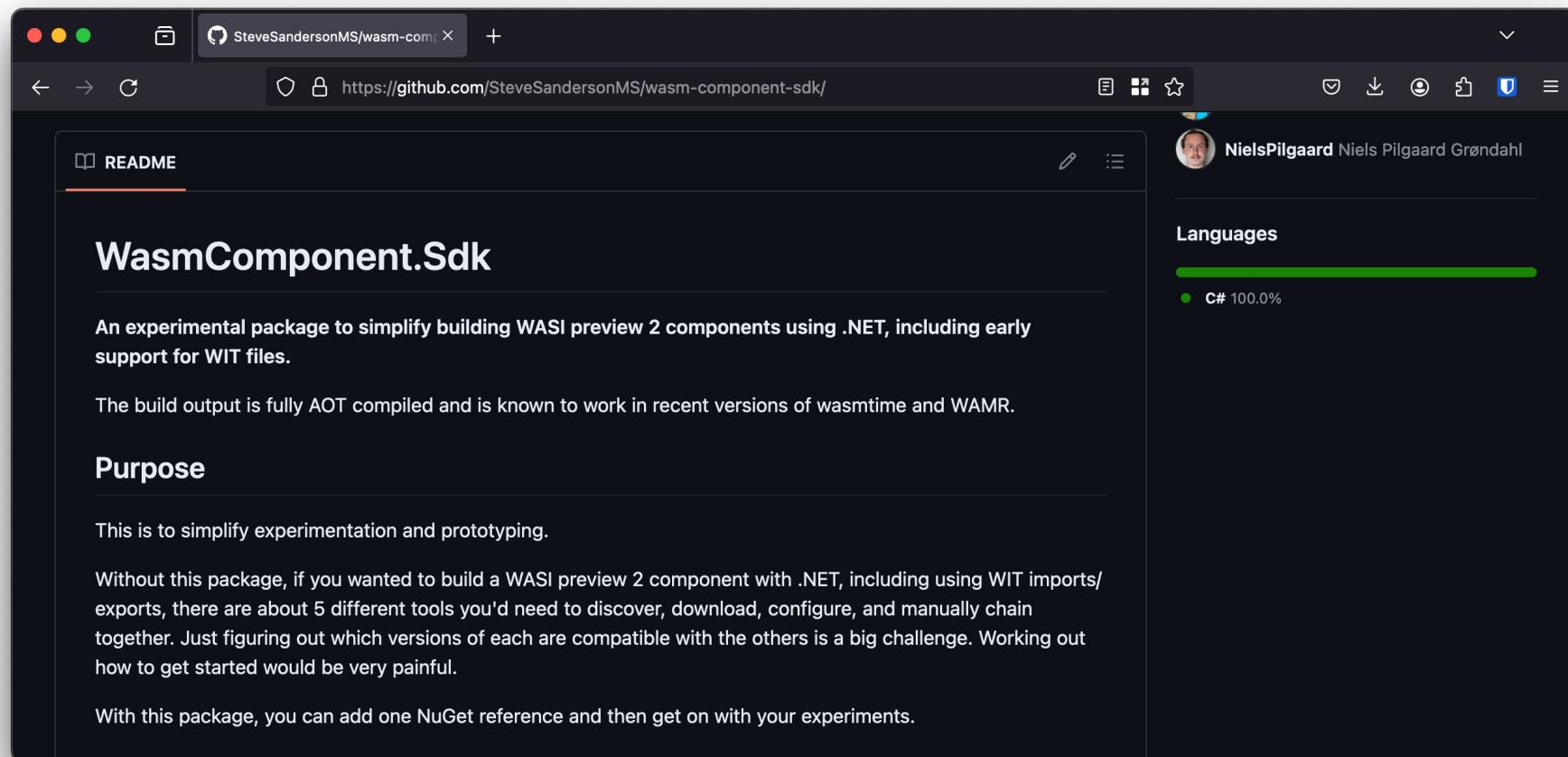
Most immediately, what this means is that the WASI Subgroup officially says that the Preview 2 APIs are stable. There is still a lot more to do, in writing more documentation, more tests, more toolchains, more implementations, and there are a lot more features that we all want to add. This vote today is a milestone along the way, rather than a destination in itself.

It also means that WASI is now officially based on the Wasm [component model](#), which makes it cross-language and virtualizable. Figuring out what a component model even is, designing it, implementing it, and building APIs using it has been a

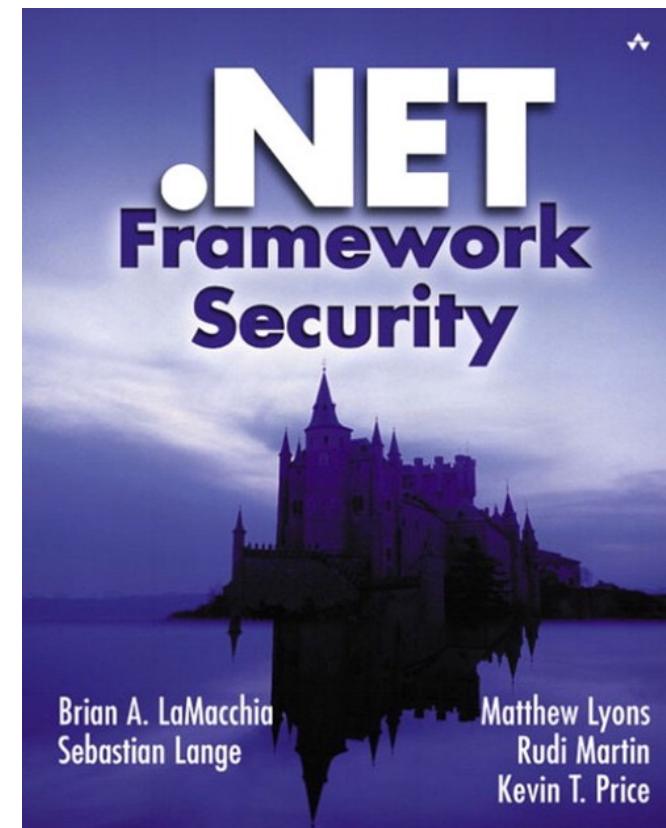
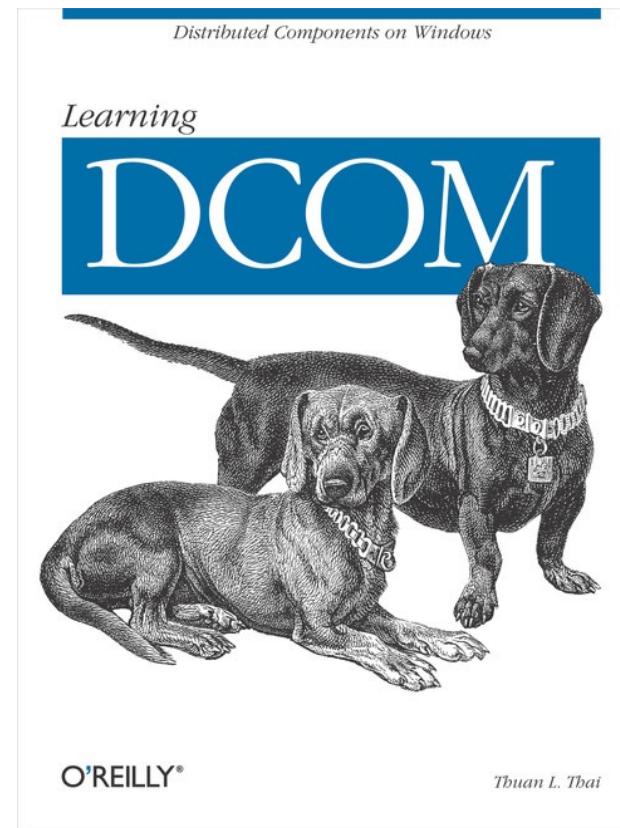
WASI Virt



WasmComponent.SDK



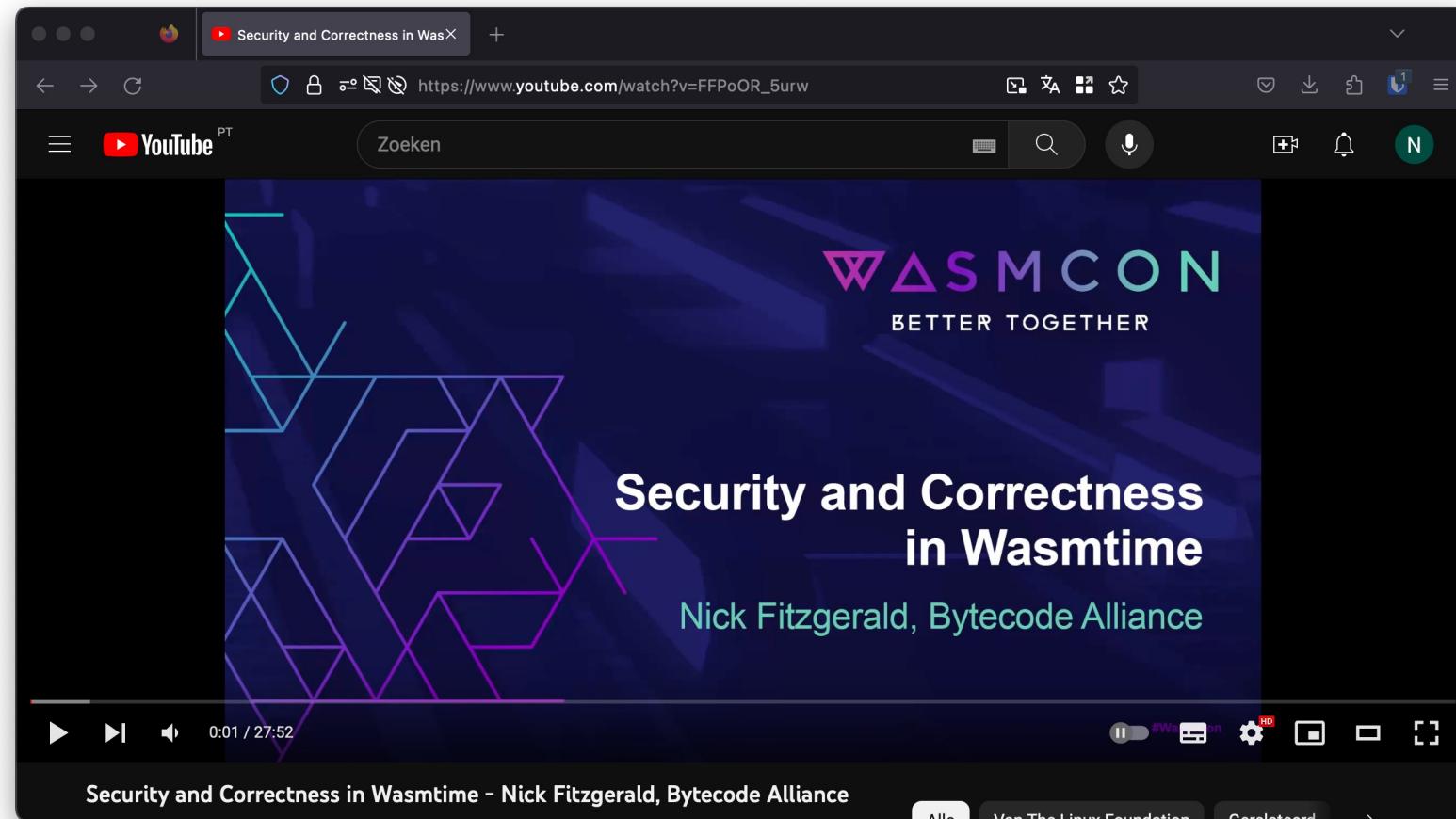
Have we seen this before?



Runtimes and Security

- Most security research published focusses on correctness of WASM runtimes/VM's
- Bytecode Alliance Blogpost September 2022:
 - "Security and Correctness in Wasmtime"
 - Written in Rust → Using all it's LangSec features
 - Continues Fuzzing & formal verification
 - Security process & vulnerability disclosure

Runtimes and Security



Conclusion

- Cloud Native ❤️ WebAssembly
- WebAssembly has a lot of potential to be used to run, extend, and secure your applications!
- Its as secure as the WebAssembly runtime implementation!
- WASI Preview 2 big milestone; now tooling can be implemented!

Questions?

- <https://github.com/nielstanis/elevatedev24wasm/>
- ntanis at Veracode.com
- @nielstanis@infosec.exchange
- <https://blog.fennec.dev>
- Merci! Thank you!