Using WebAssembly to run, extend, and secure your .NET/Java/Python/Go/… application

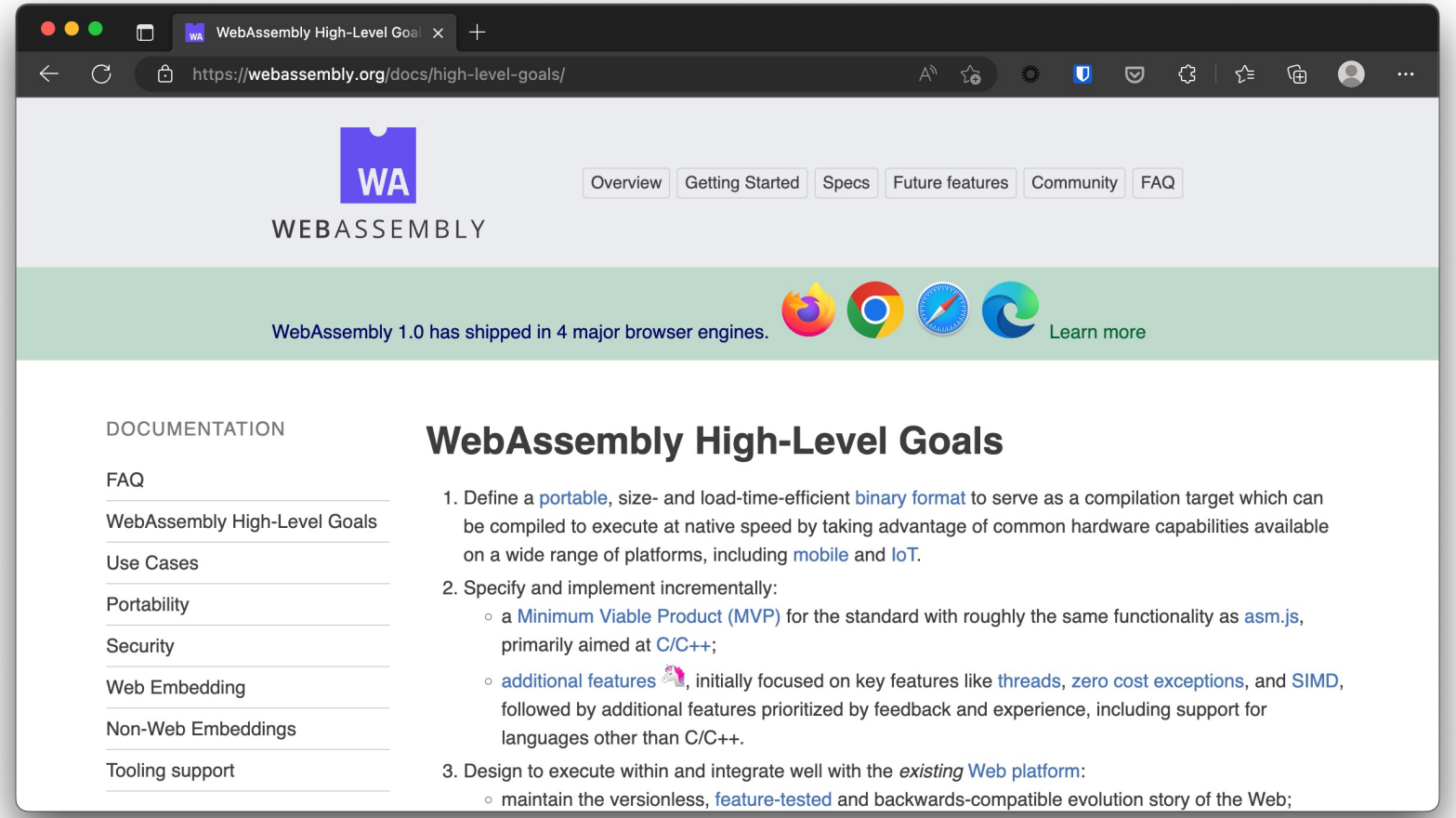
Niels Tanis

Who am I?

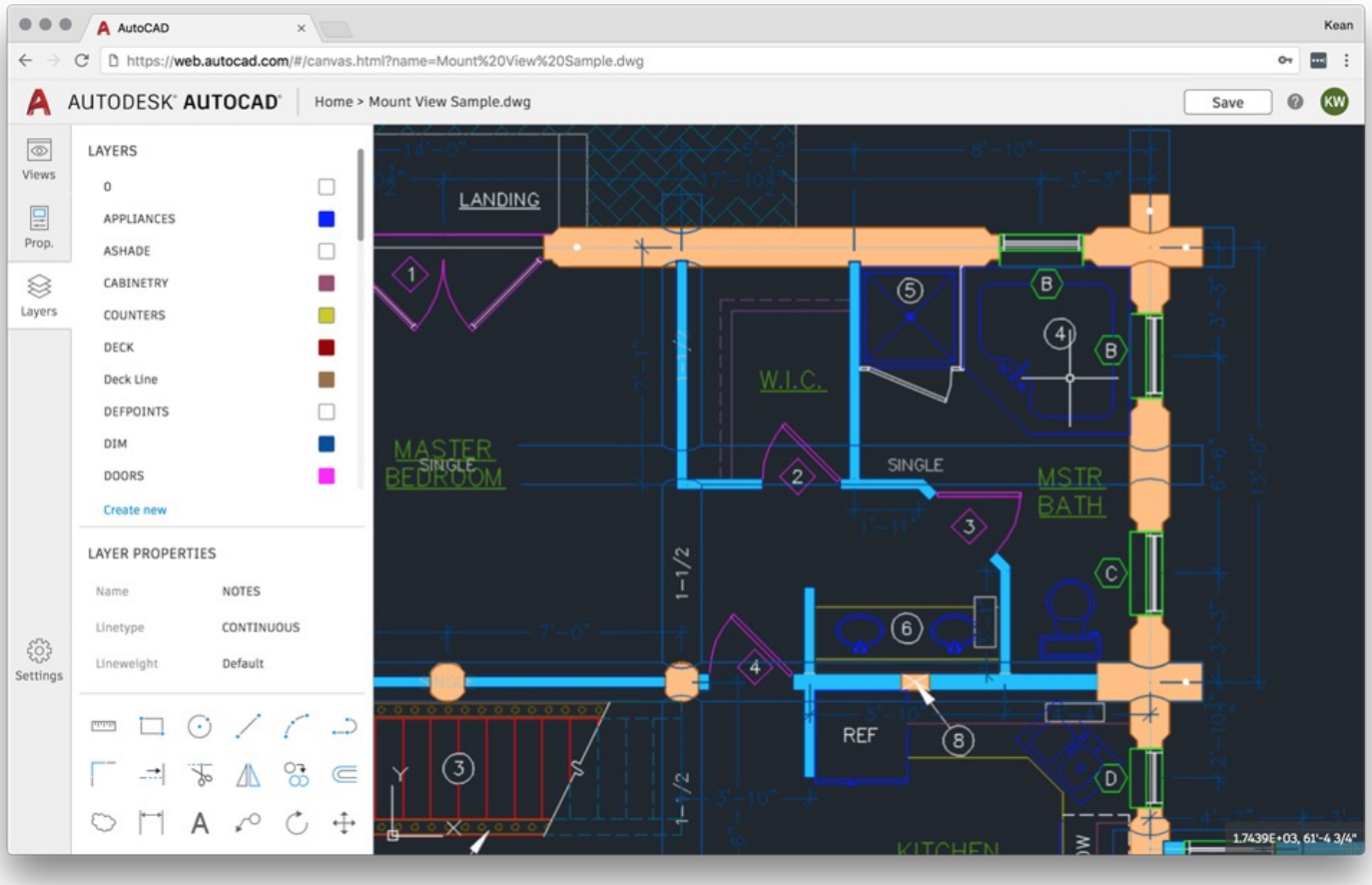
- Niels Tanis
- Sr. Principal Security Researcher
  - Background .NET Development, Pentesting/ethical hacking, and software security consultancy
  - Research on static analysis for .NET apps
- Enjoying programming in Rust!
- Microsoft MVP – Developer Technologies

@nielstanis@infosec.exchange

WebAssembly



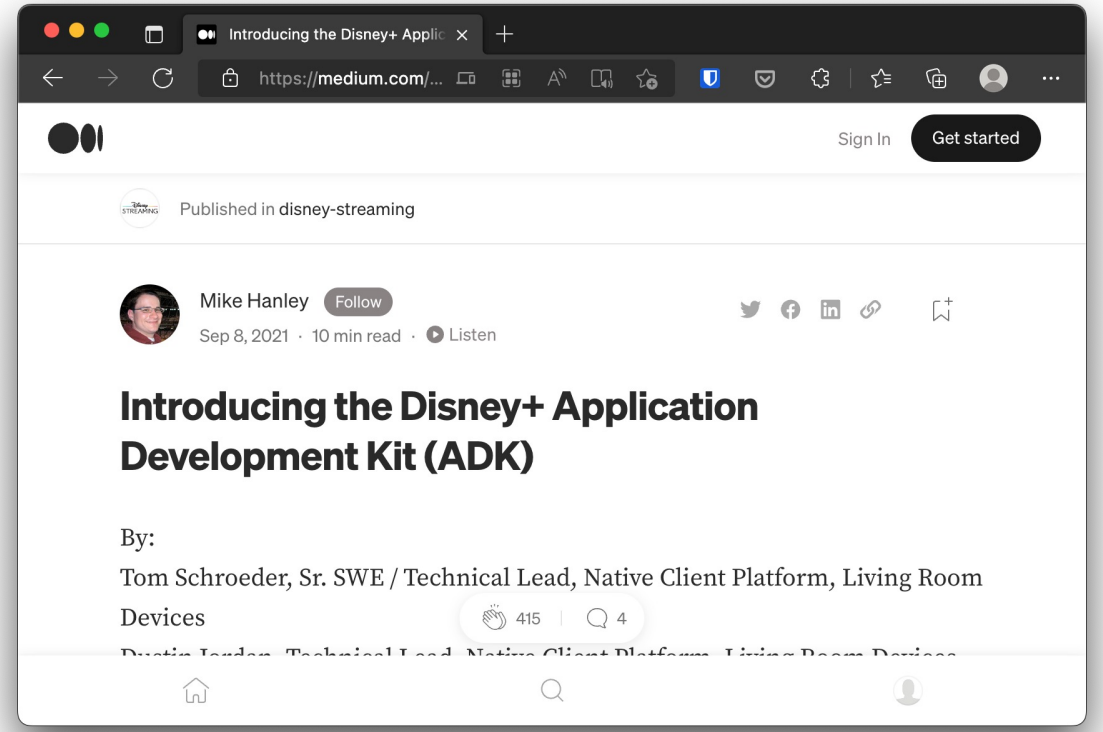
WebAssembly - AutoCAD

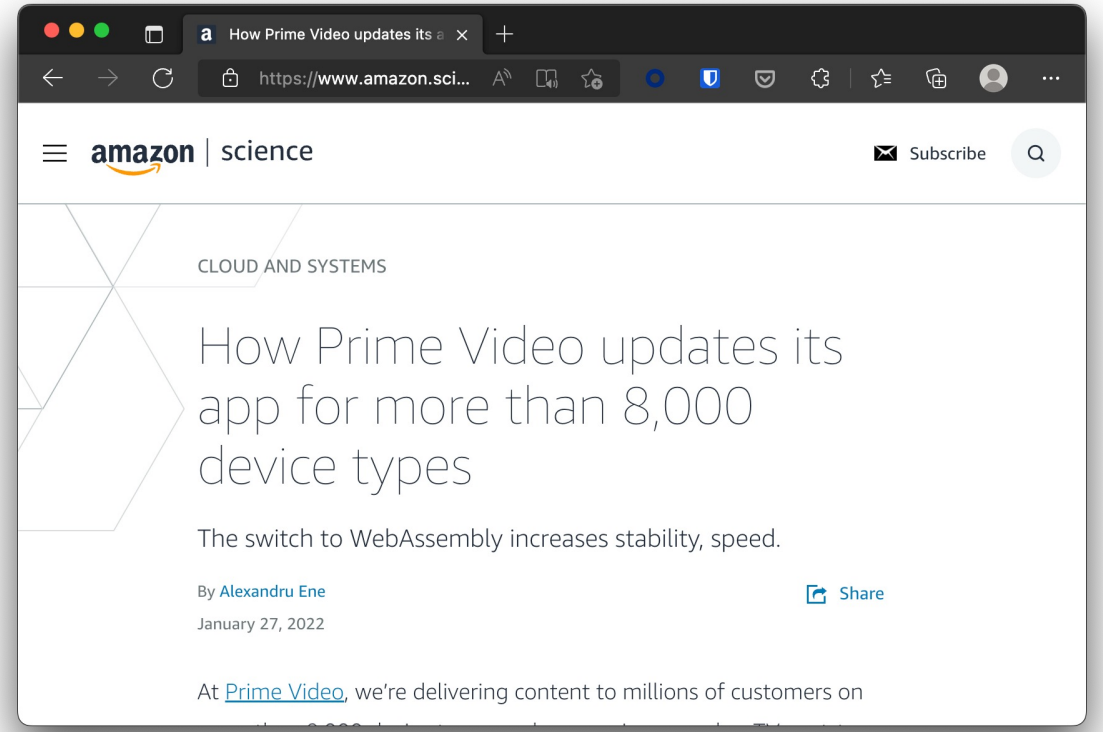


CODE EUROPE

@nielstanis@infosec.exchange

WebAssembly - SDK's





Agenda

- Introduction
- WebAssembly 101
- Running .NET on WebAssembly
- Extending .NET with WebAssembly
- Securing .NET with WebAssembly
- Conclusion
- Q&A

WebAssembly Design

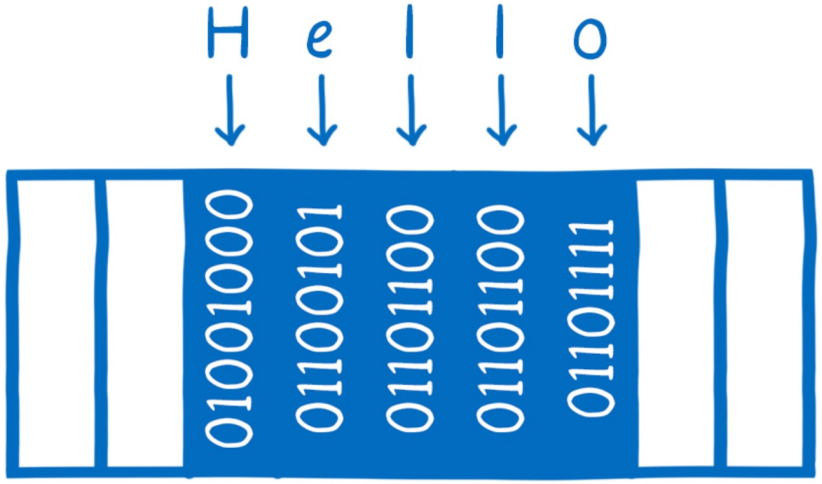
- **Be fast, efficient, and portable**
  - Executed in near-native speed across different platforms
- **Be readable and debuggable**
  - In low-level bytecode but also human readable
- **Keep secure**
  - Run on sandboxed execution environment
- **Don't break the web**
  - Ensure backwards compatibility

WebAssembly

- Binary instruction format for stack-based virtual machine similar to .NET running MSIL
- Designed as a portable compilation target
- The security model of WebAssembly:
  - Protect users from buggy or malicious modules
  - Provide developers with useful primitives and mitigations for developing safe applications

WebAssembly Memory

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

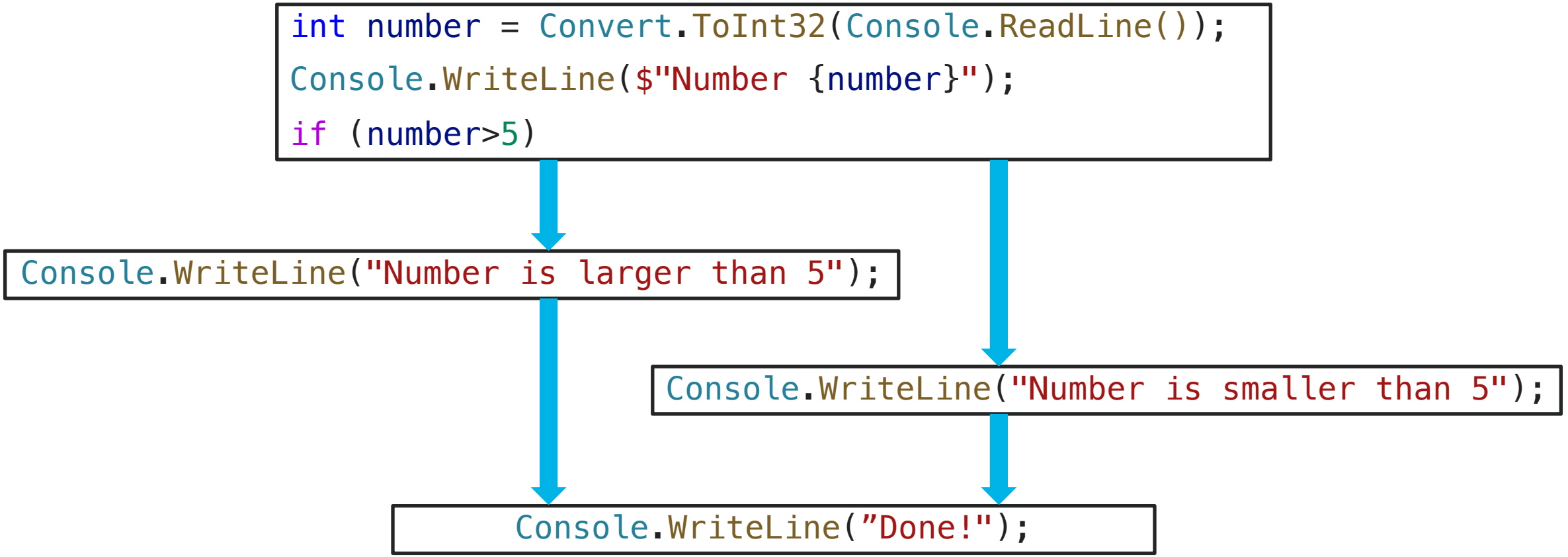


@nielstanis@infosec.exchange

WebAssembly Control-Flow Integrity

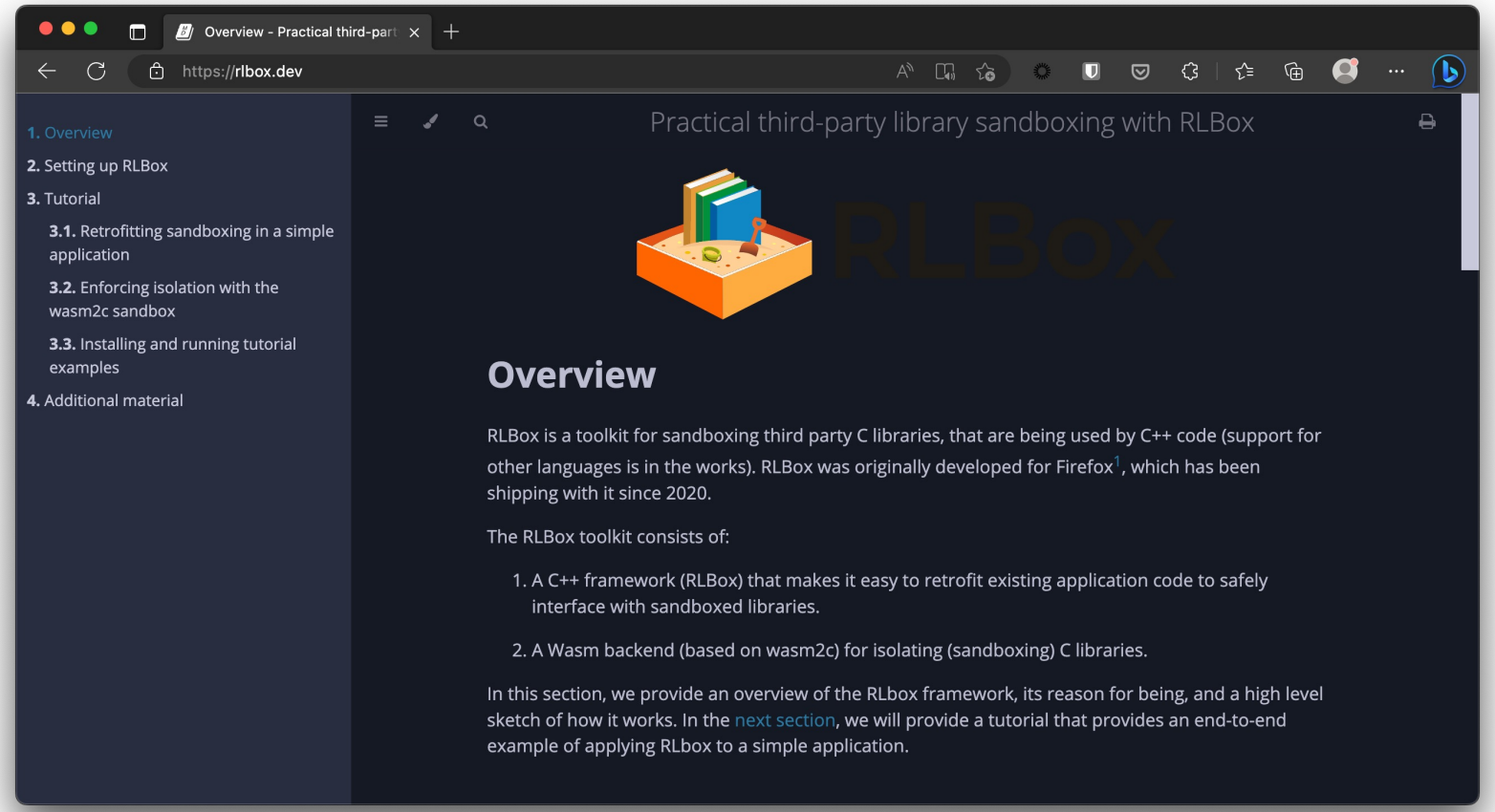
```
int number = Convert.ToInt32(Console.ReadLine());
Console.WriteLine($"Number {number}");
if (number>5)
{
        Console.WriteLine("Number is larger than 5");
}
else
{
        Console.WriteLine("Number is smaller than 5");
}
Console.WriteLine(”Done!");
```

WebAssembly Control-Flow Integrity



@nielstanis@infosec.exchange

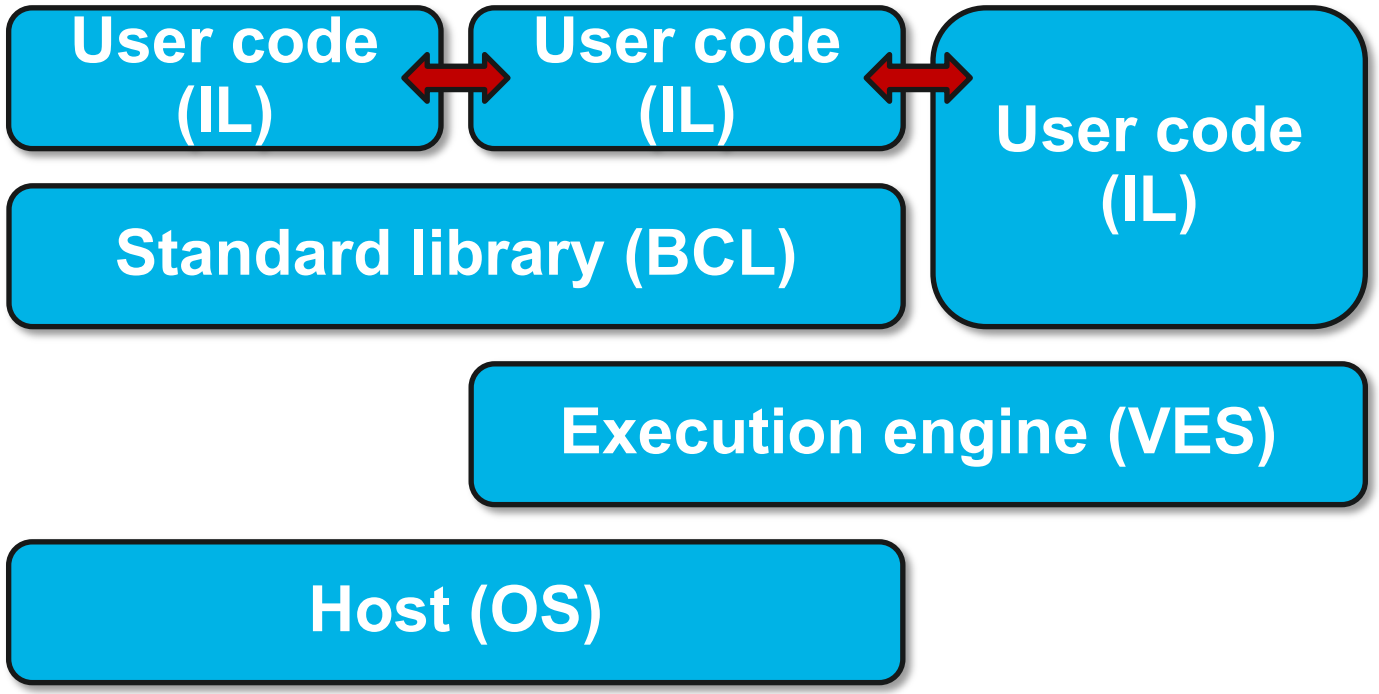
FireFox RLBox



CODE EUROPE

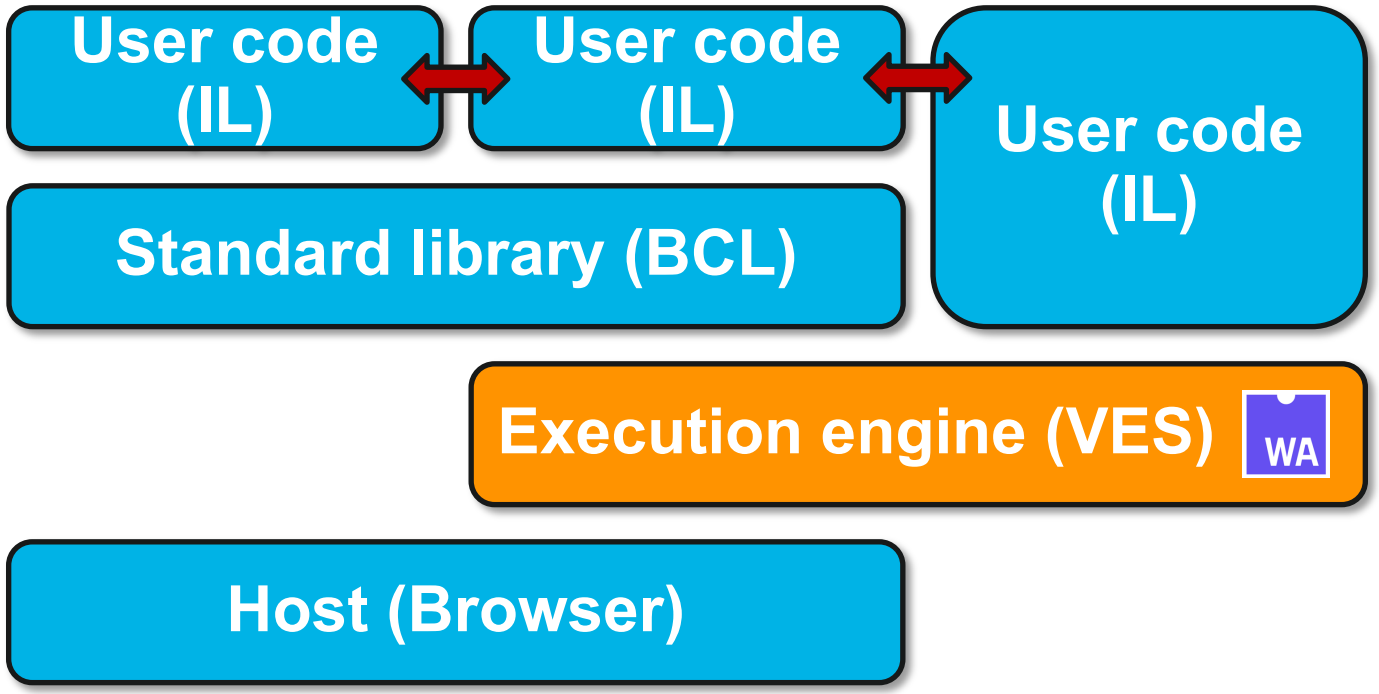
@nielstanis@infosec.exchange

Running .NET on WebAssembly

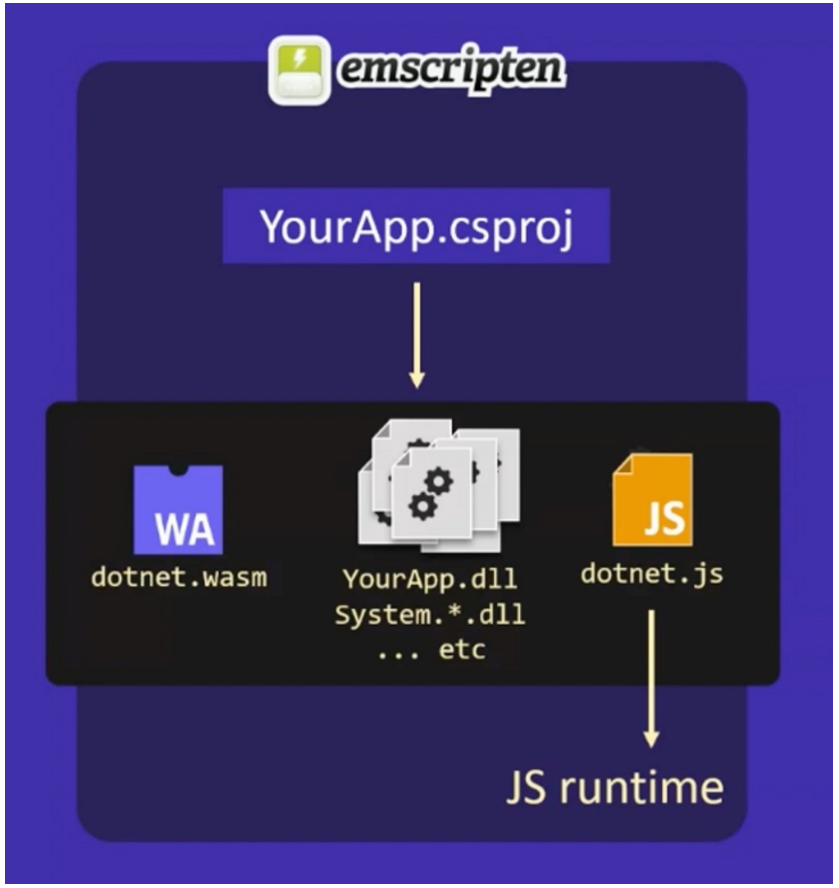


@nielstanis@infosec.exchange

Running .NET on WebAssembly

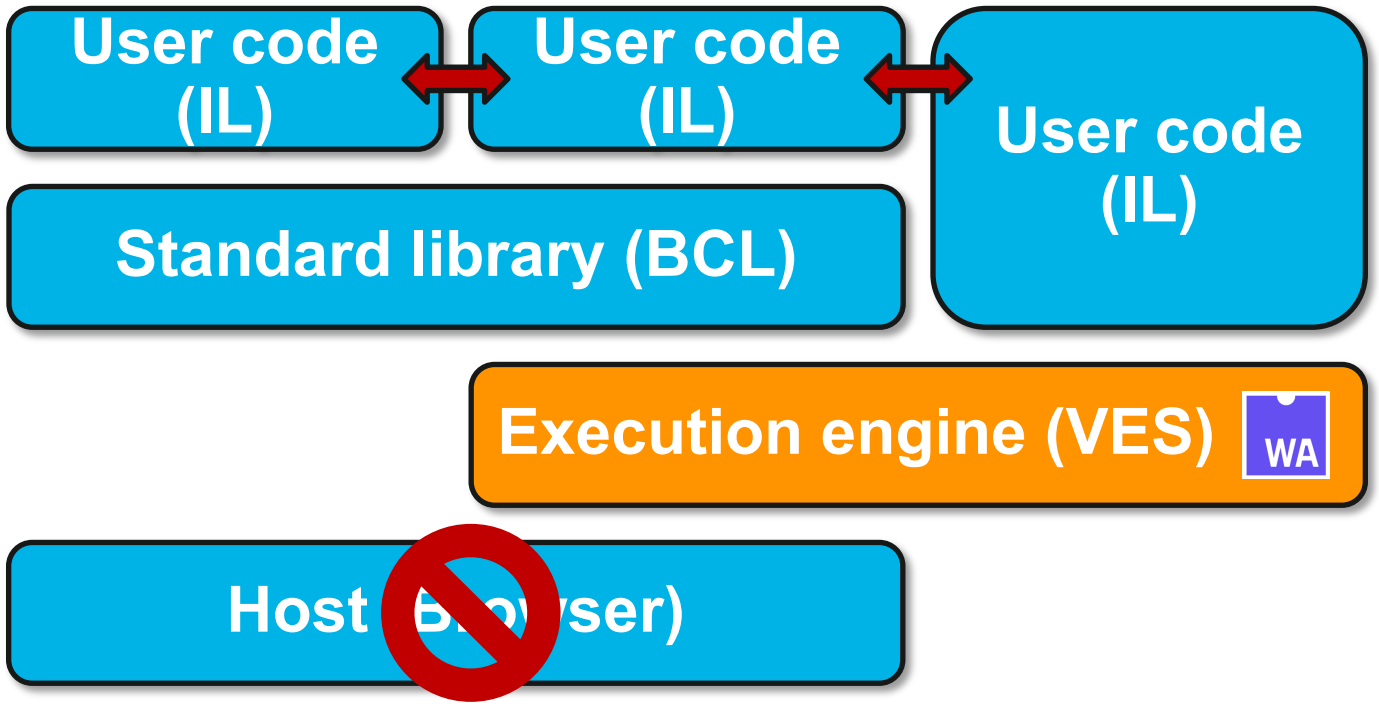


Blazor WebAssembly



@nielstanis@infosec.exchange

Running .NET on WebAssembly



@nielstanis@infosec.exchange

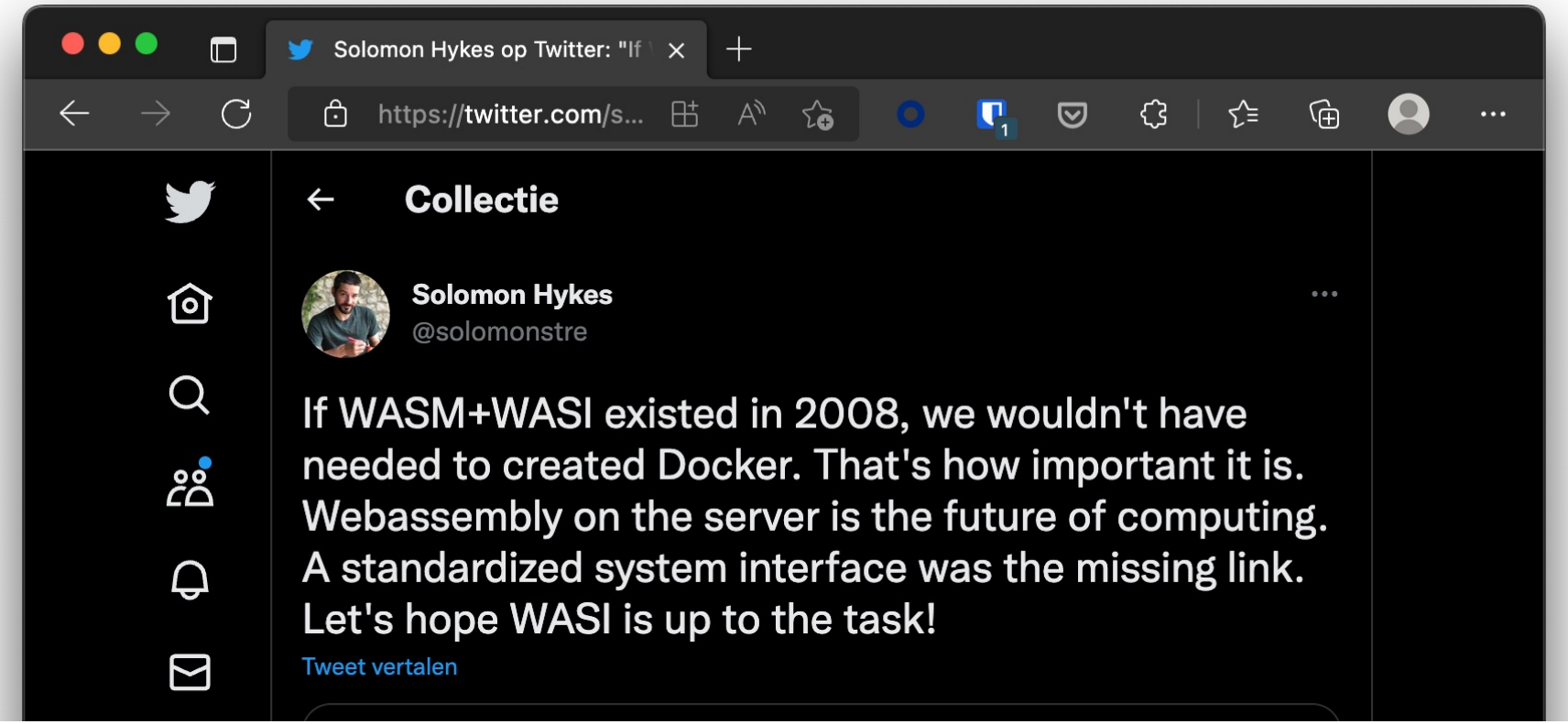
WebAssembly System Interface WASI

- 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
- Right now, supports e.g. FileSystem actions
- Future support for sockets and other system resources.
- Anyone recall .NET Standard? ☺

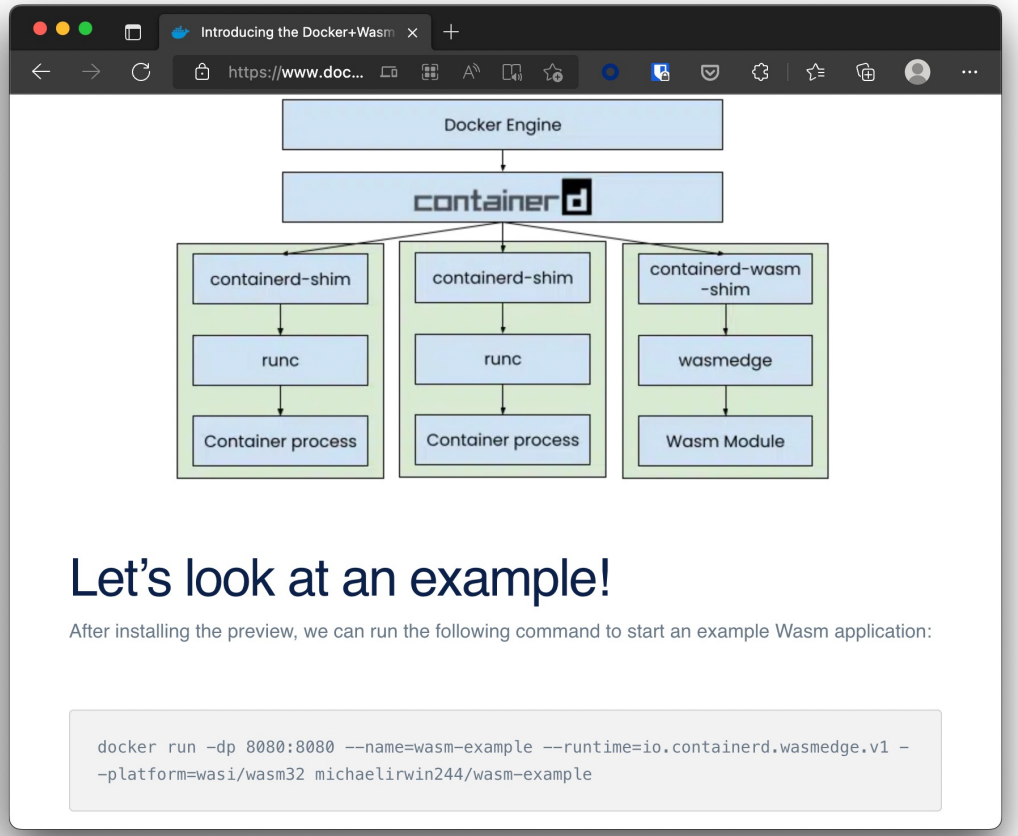
Docker vs WASM & WASI



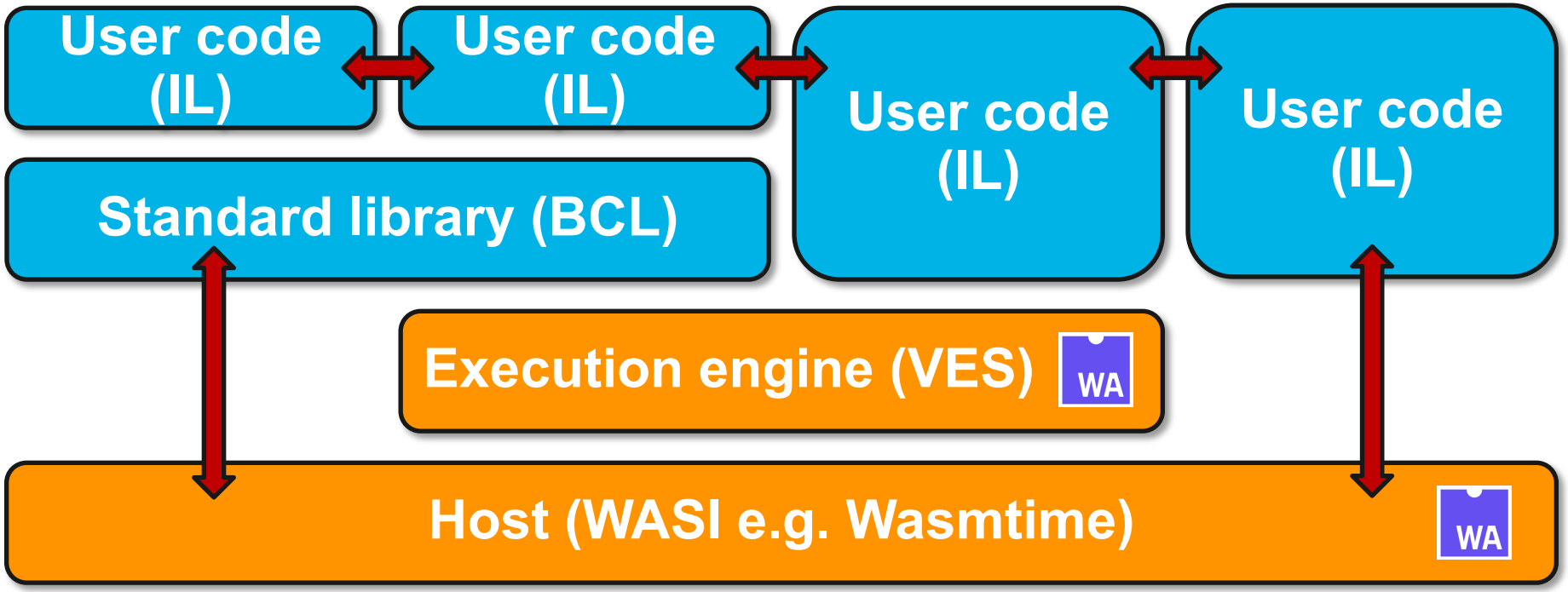
Docker vs WASM & WASI

Docker & WASM


Introducing the Docker+Wasm
https://www.doc...
Wasm is a fast, light alternative to Linux containers — try it out today in the Docker+Wasm Technical Preview
docker +WASM
Introducing 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.

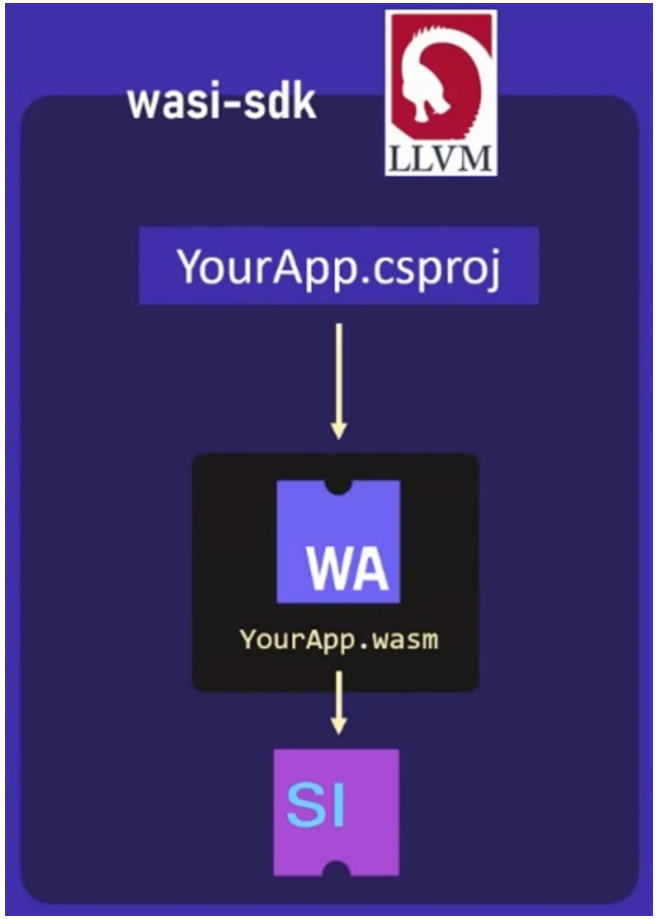

Introducing the Docker+Wasm
https://www.doc...
Docker Engine
containerd
containerd-shim
runc
Container process
containerd-shim
runc
Container process
containerd-wasm -shim
wasmedge
Wasm Module
Let's look at an example!
After installing the preview, we can run the following command to start an example Wasm application:
docker run -dp 8080:8080 --name=wasm-example --runtime=io.containerd.wasmedge.v1 --platform=wasi/wasm32 michaelirwin244/wasm-example

WebAssembly System Interface WASI



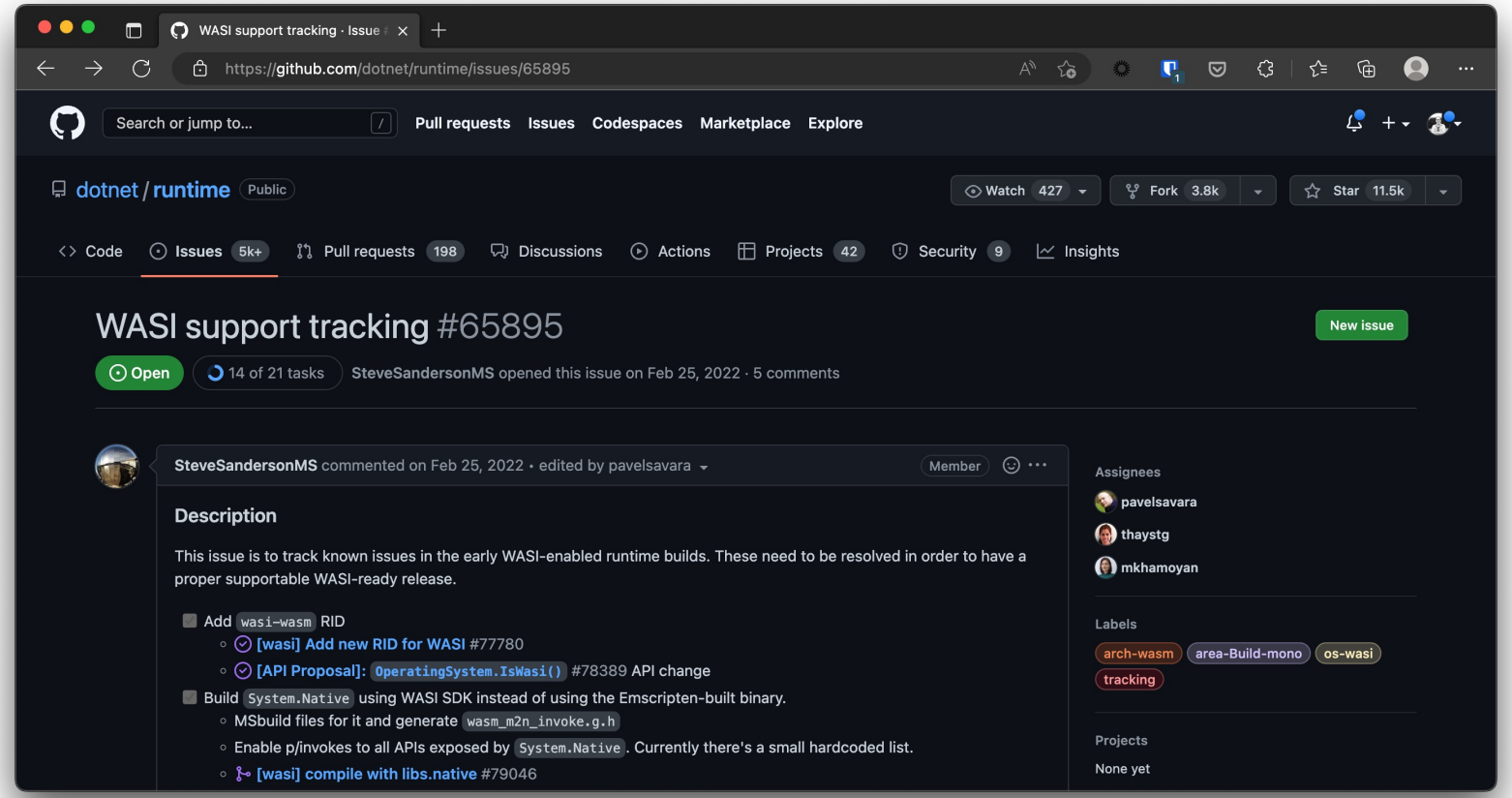
@nielstanis@infosec.exchange

Experimental WASI SDK for .NET



@nielstanis@infosec.exchange

Experimental WASI SDK for .NET



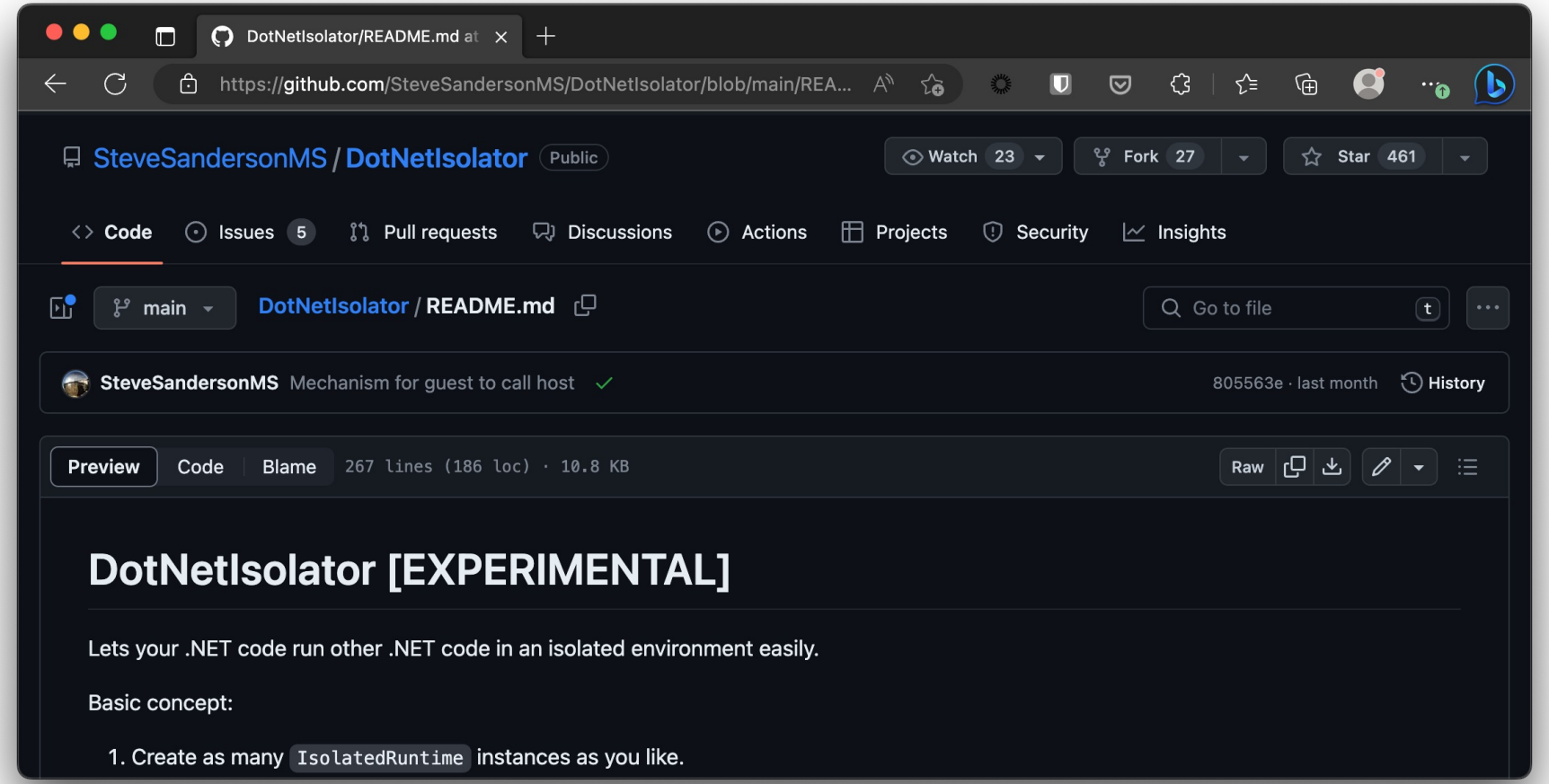
CODE EUROPE

@nielstanis@infosec.exchange

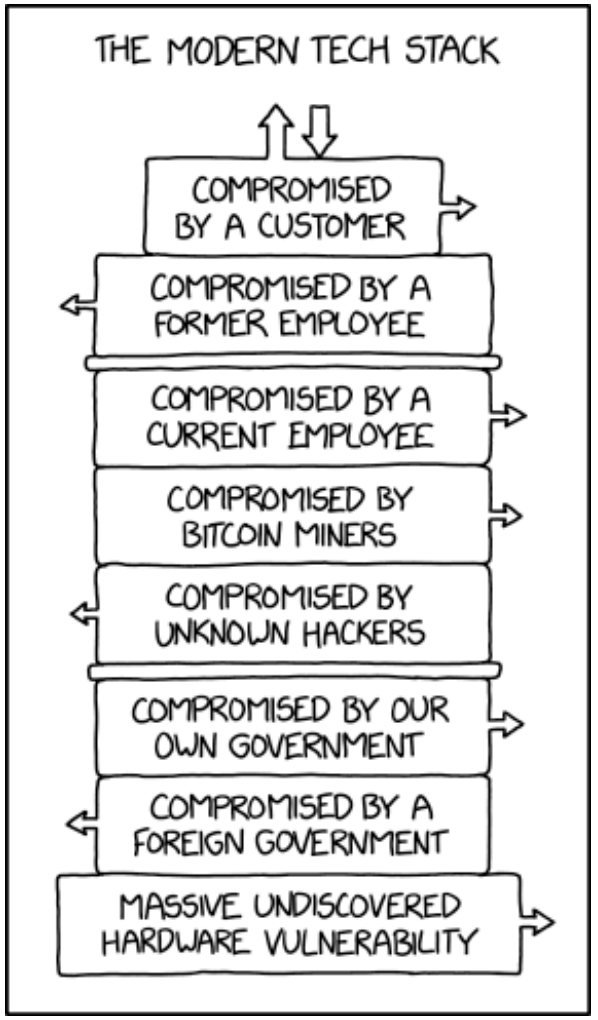
Extending .NET with WASM

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

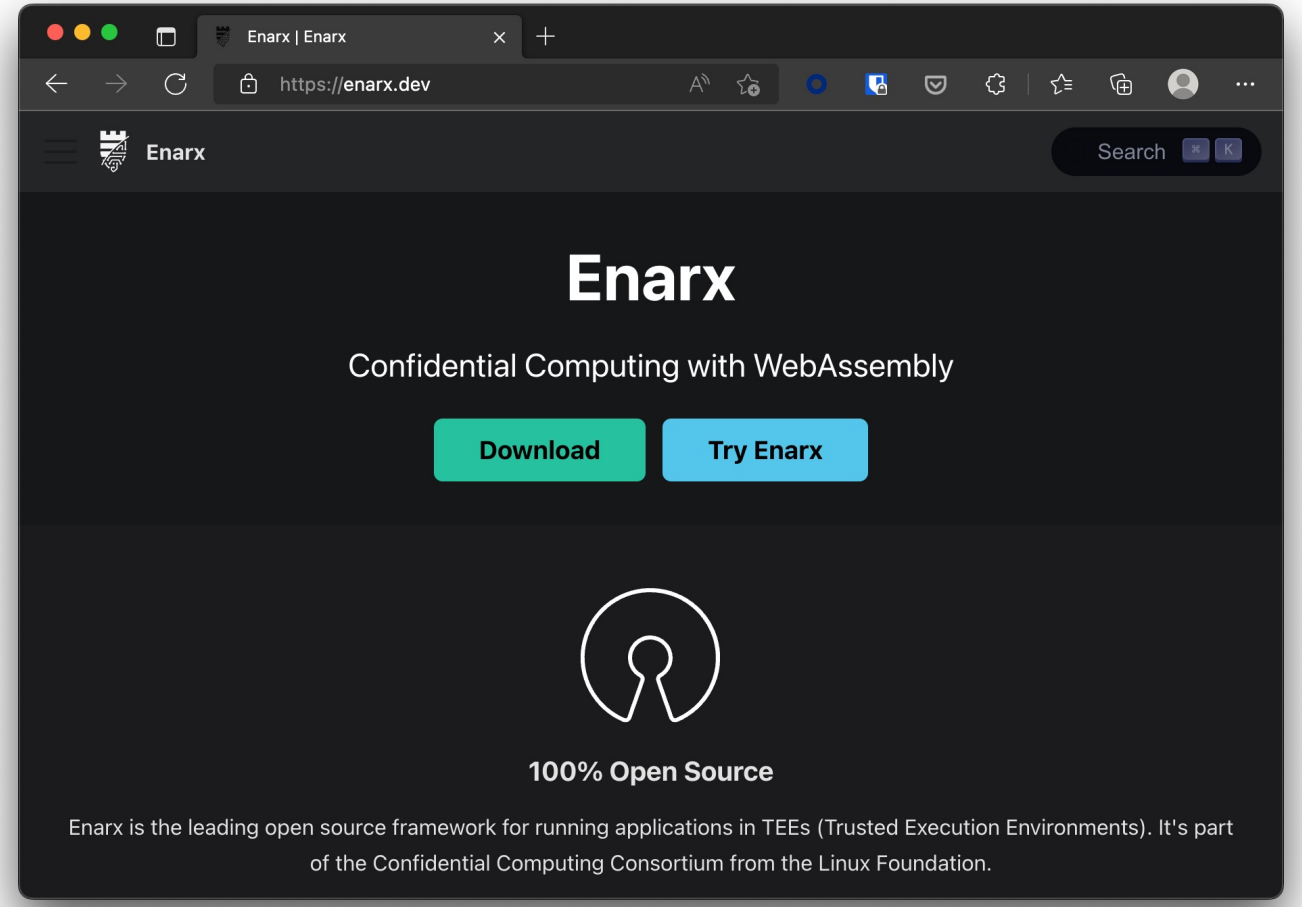
DotNetIsolator



Trusted Computing - XKCD 2166

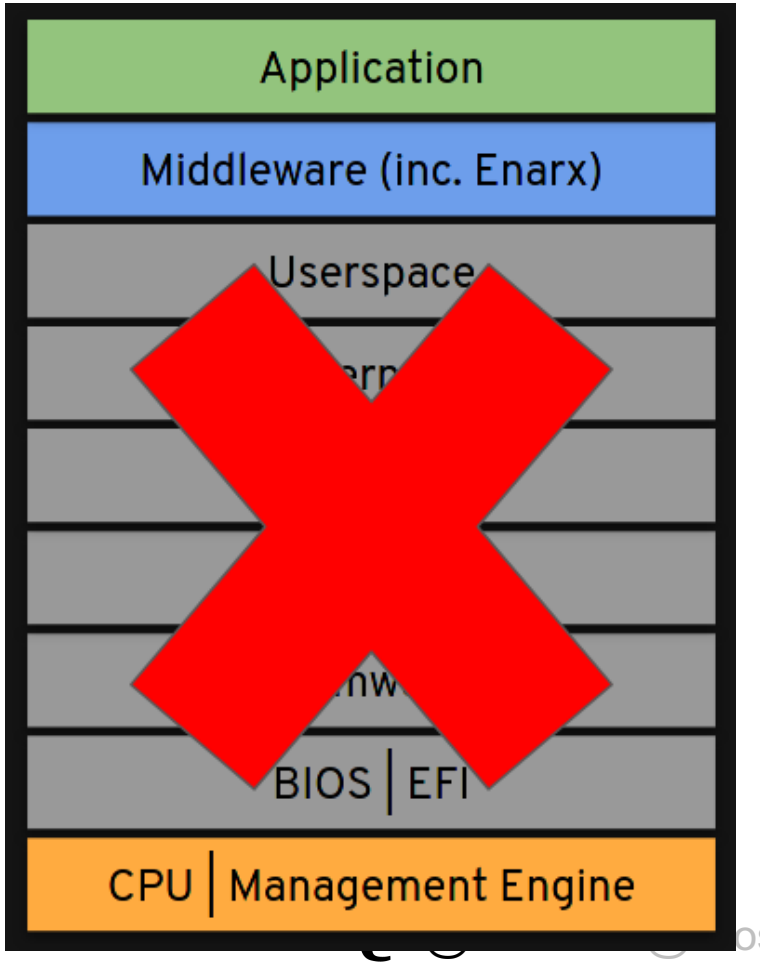


Enarx


Enarx | Enarx
https://enarx.dev
Enarx
Search
Enarx
Confidential Computing with WebAssembly
Download
Try Enarx
100% Open Source
Enarx is the leading open source framework for running applications in TEEs (Trusted Execution Environments). It's part of the Confidential Computing Consortium from the Linux Foundation.

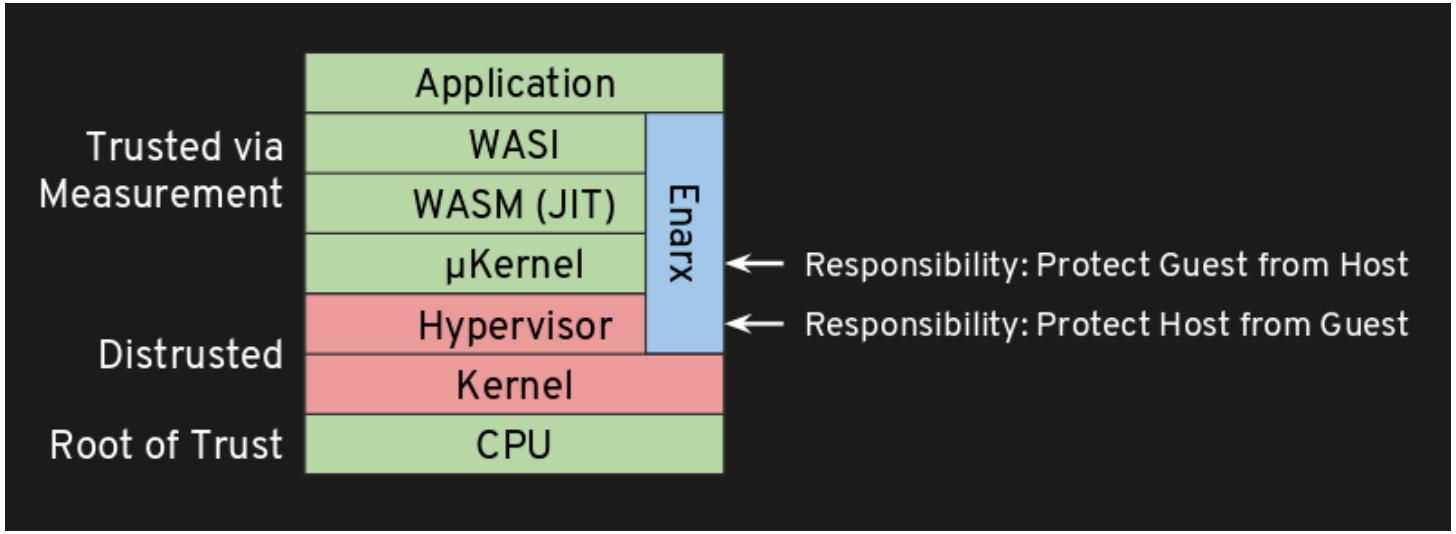
Enarx Threat Model

- 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

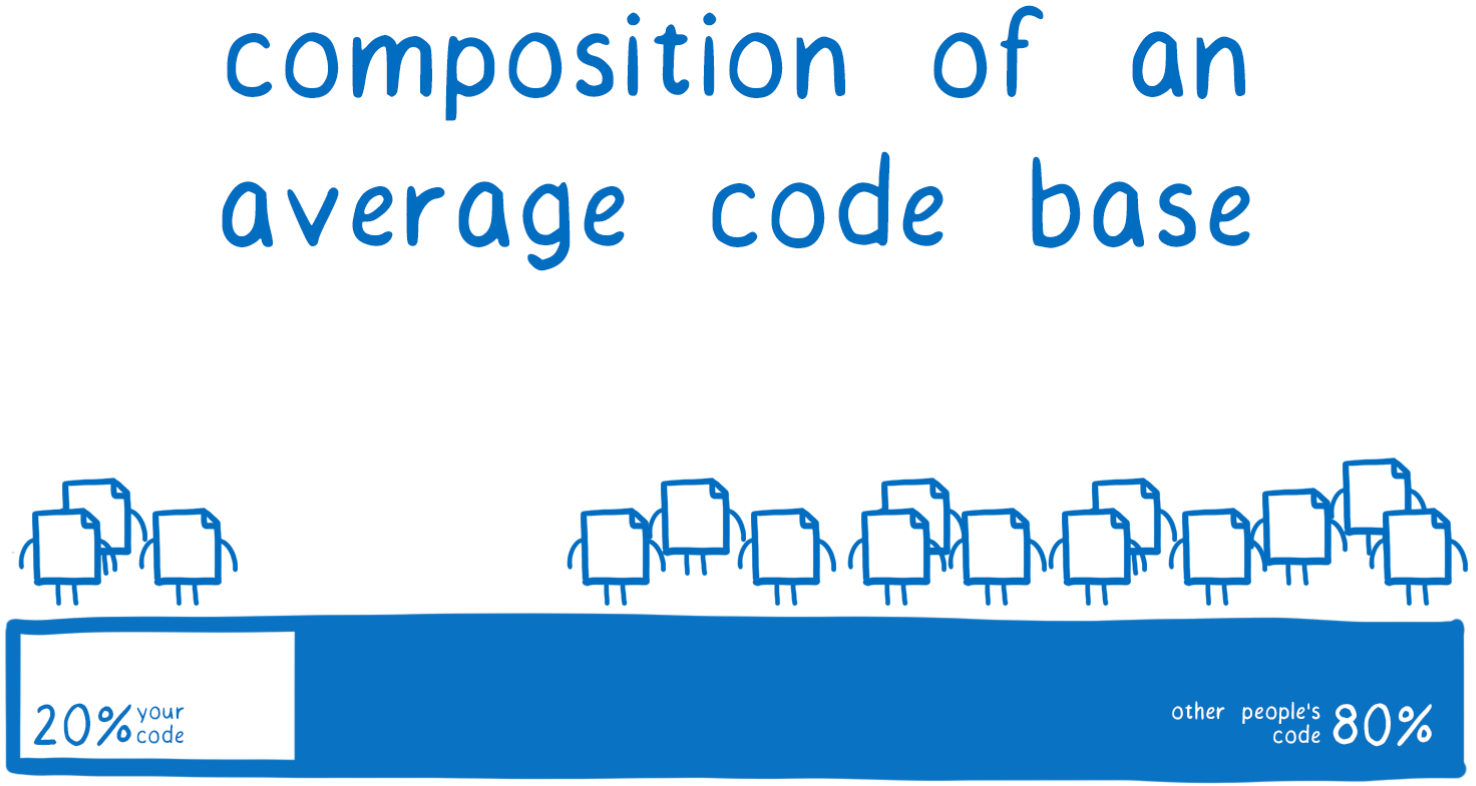


Enarx

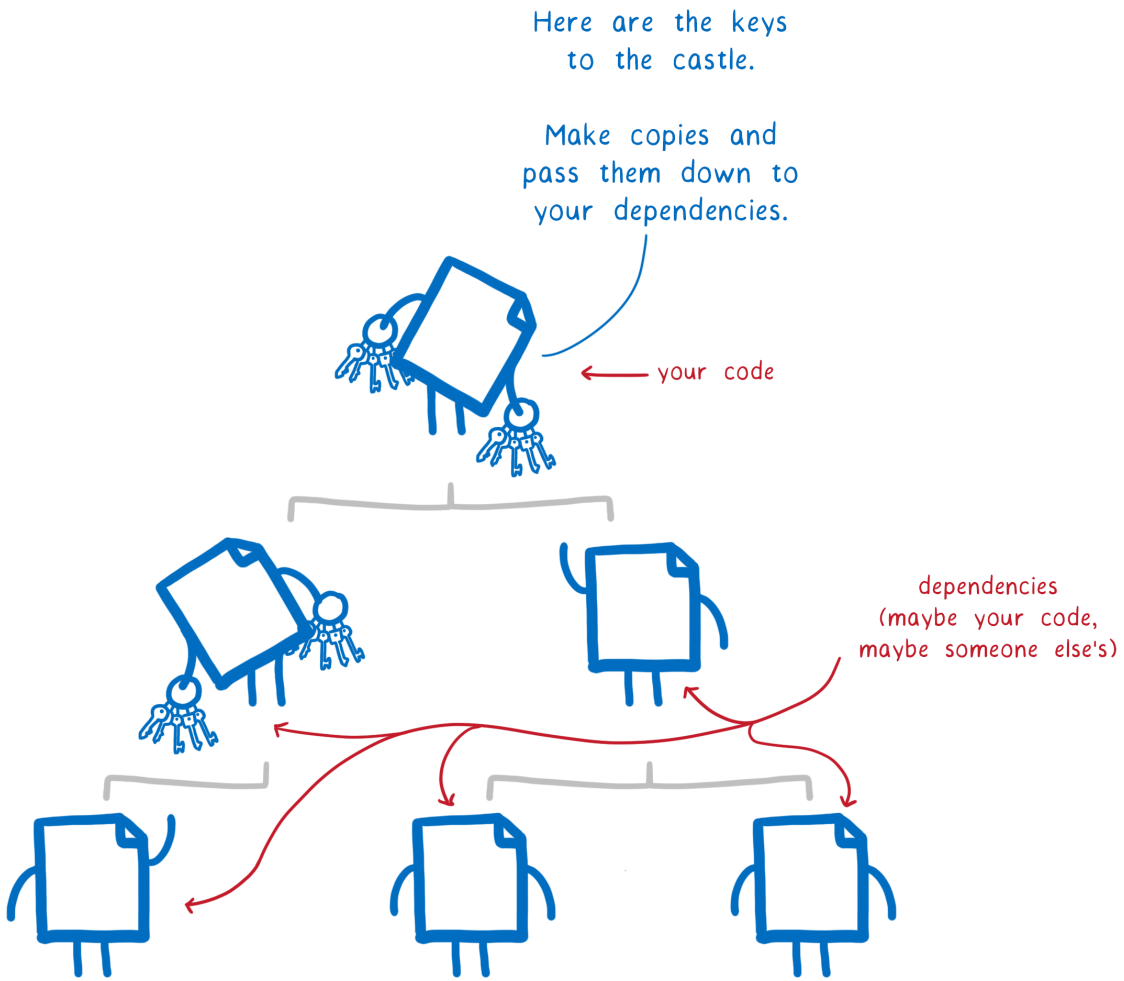
- Leverages Trusted Execution Environment (TEE) direct on processor
  - AMD's SEV, Intel's SGX and IBM's PEF
- Attestation of hardware and Enarx runtime



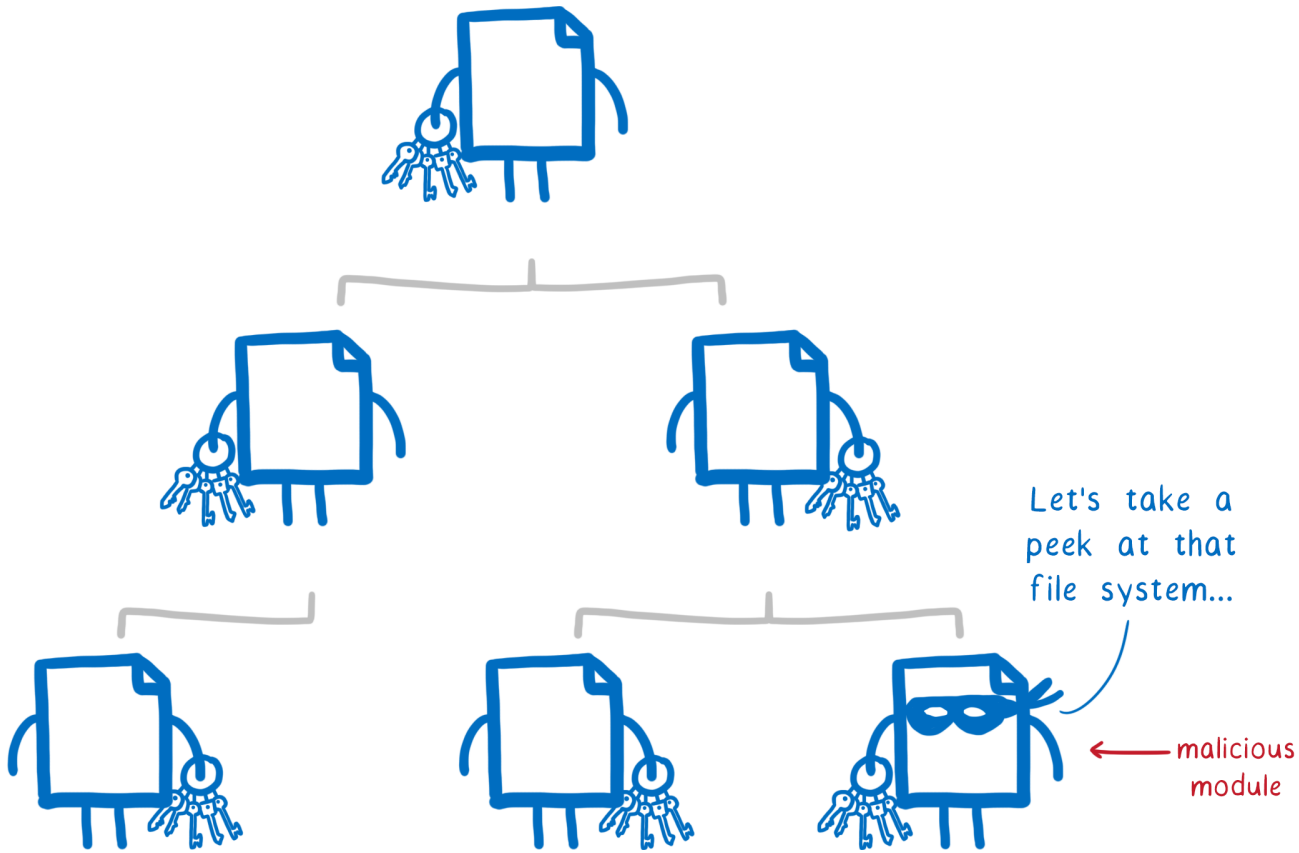
WASM – What's next?



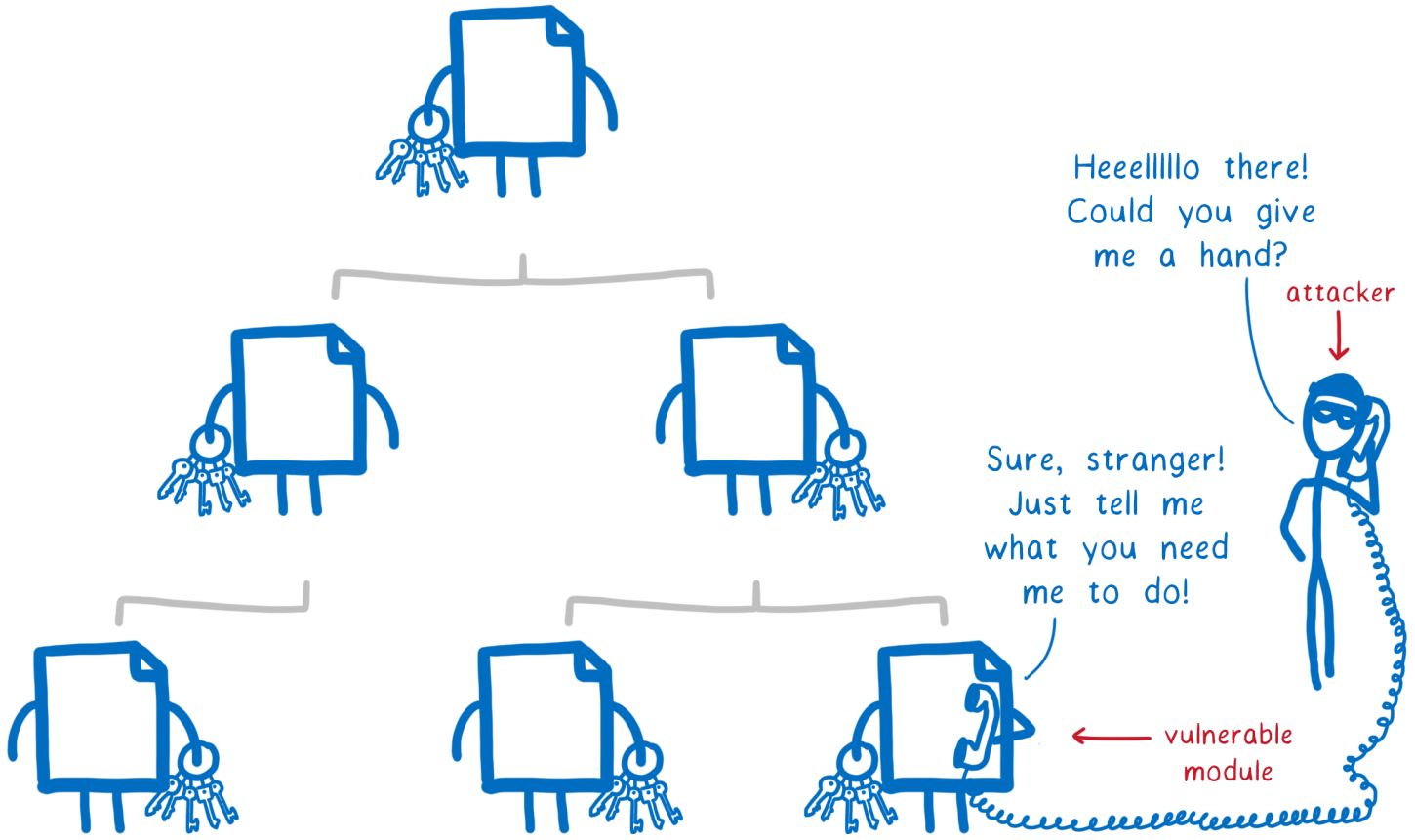
Dependencies



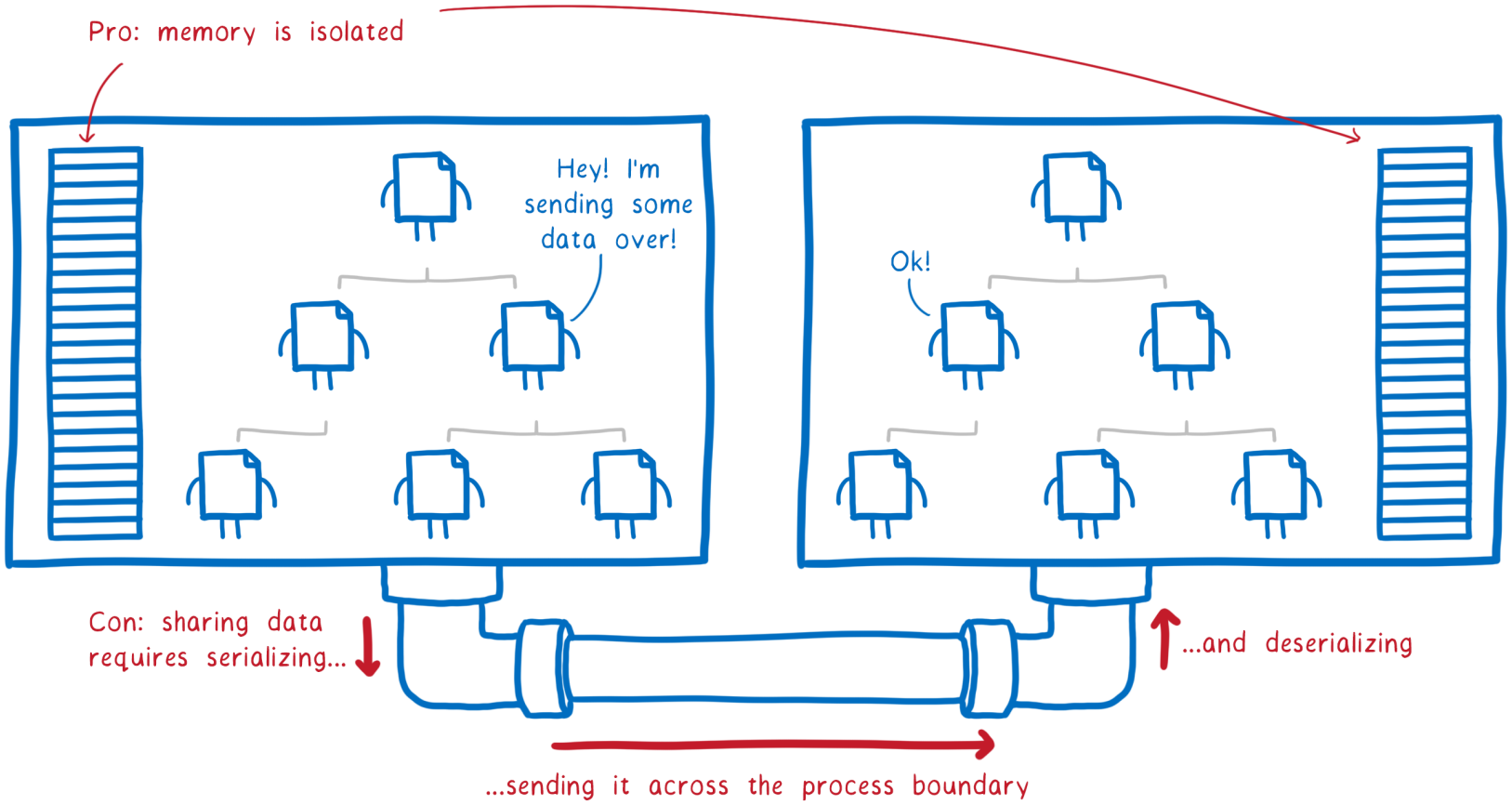
Malicious module



Vulnerable module

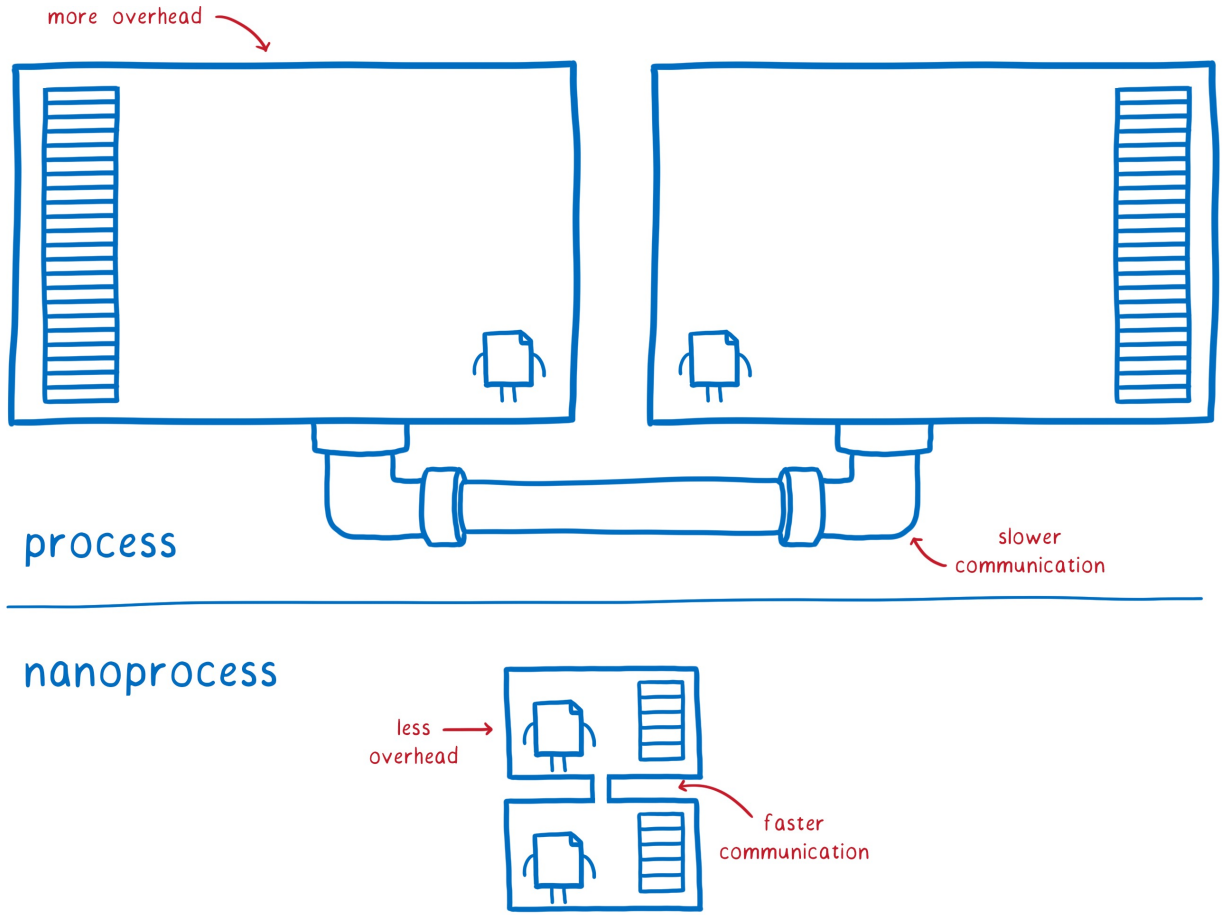


Process Isolation



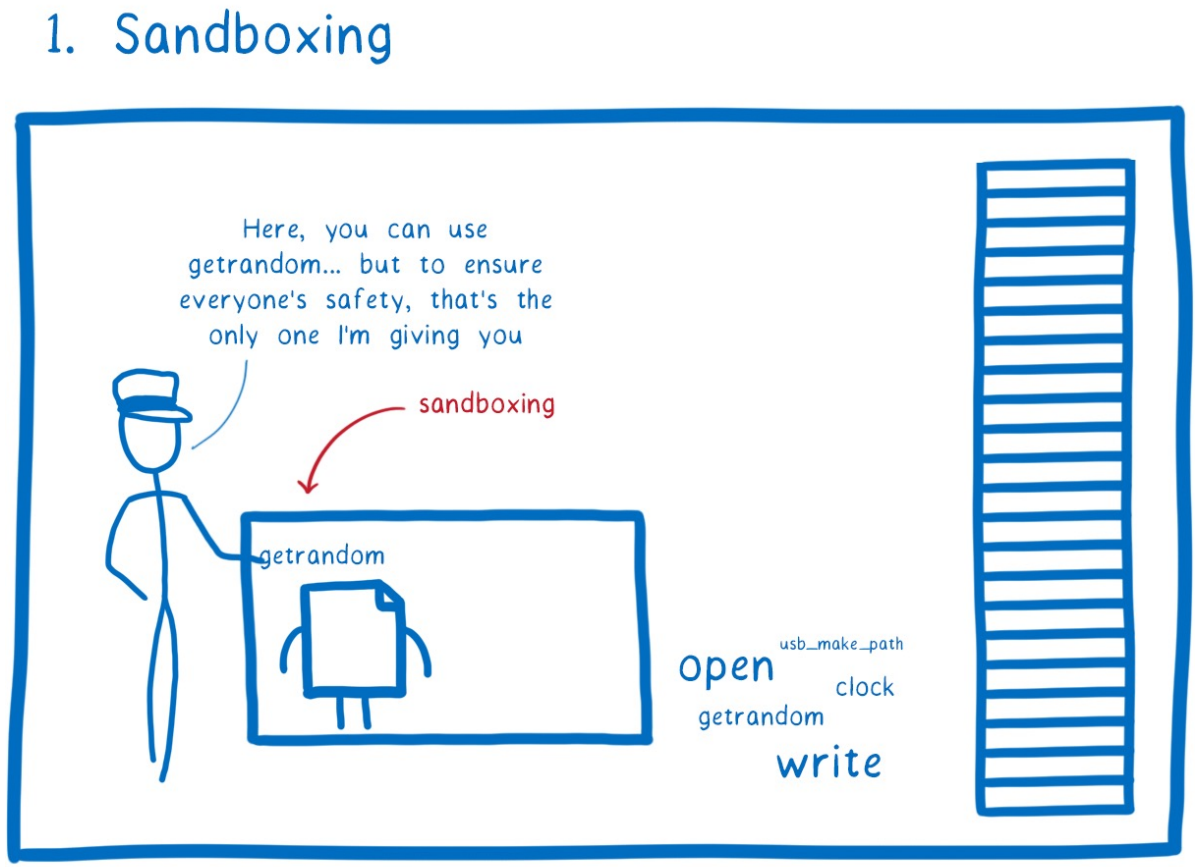
@nielstanis@infosec.exchange

WebAssembly Nano-Process

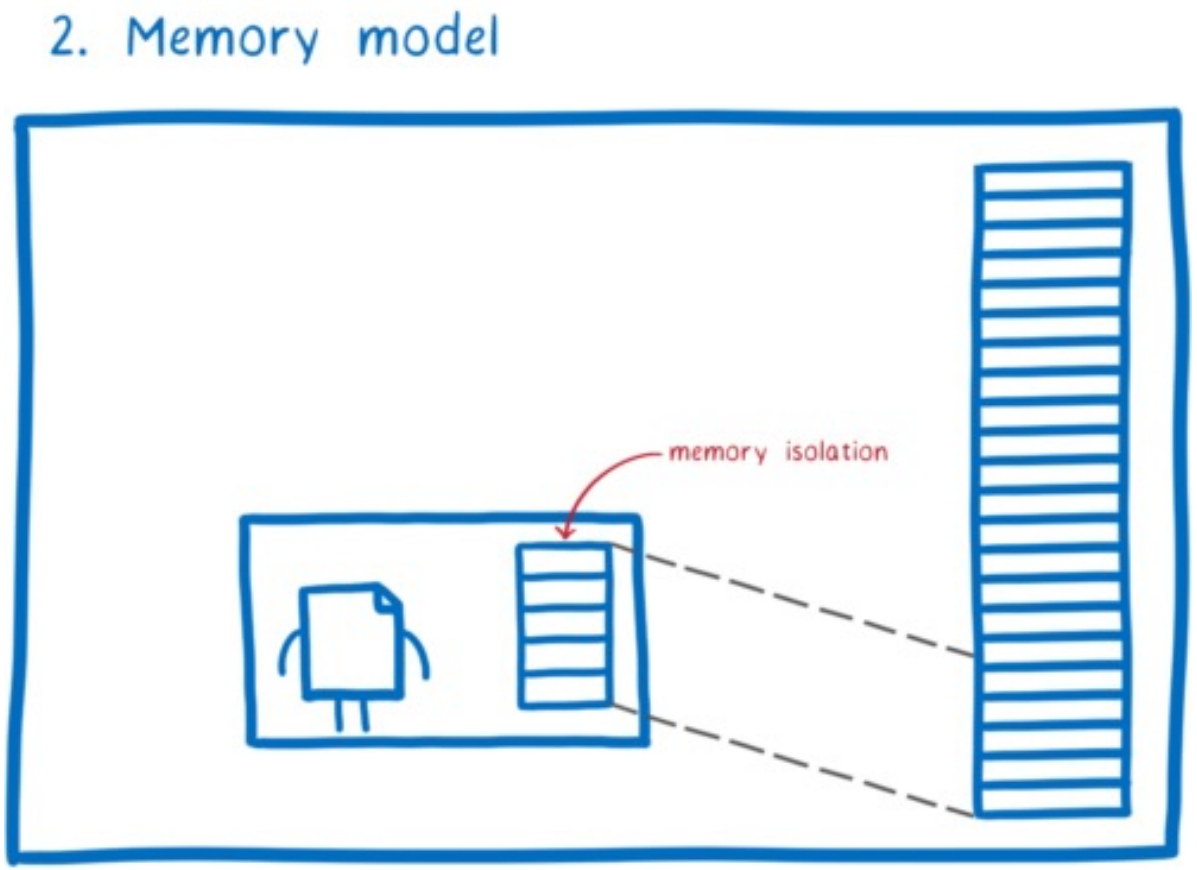


* not drawn to scale

WebAssembly Nano-Process

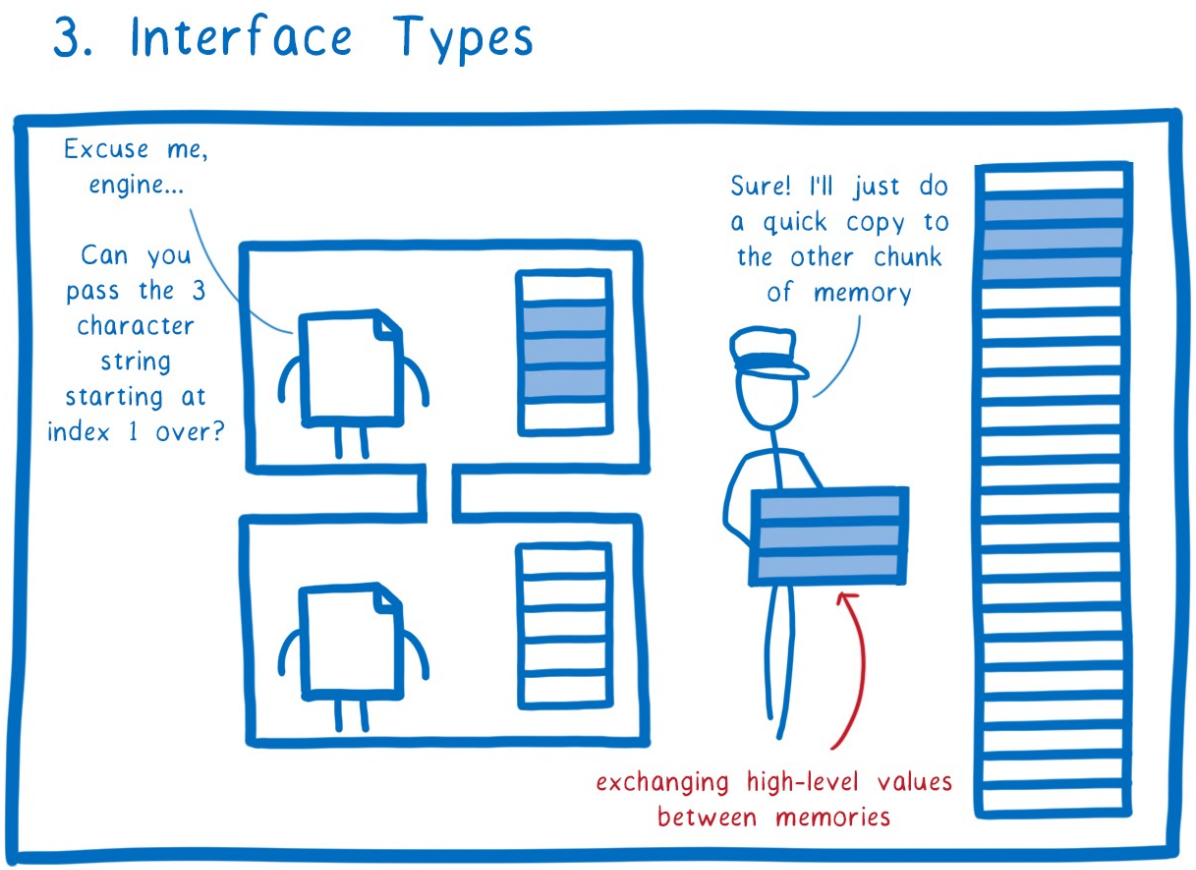


WebAssembly Nano-Process

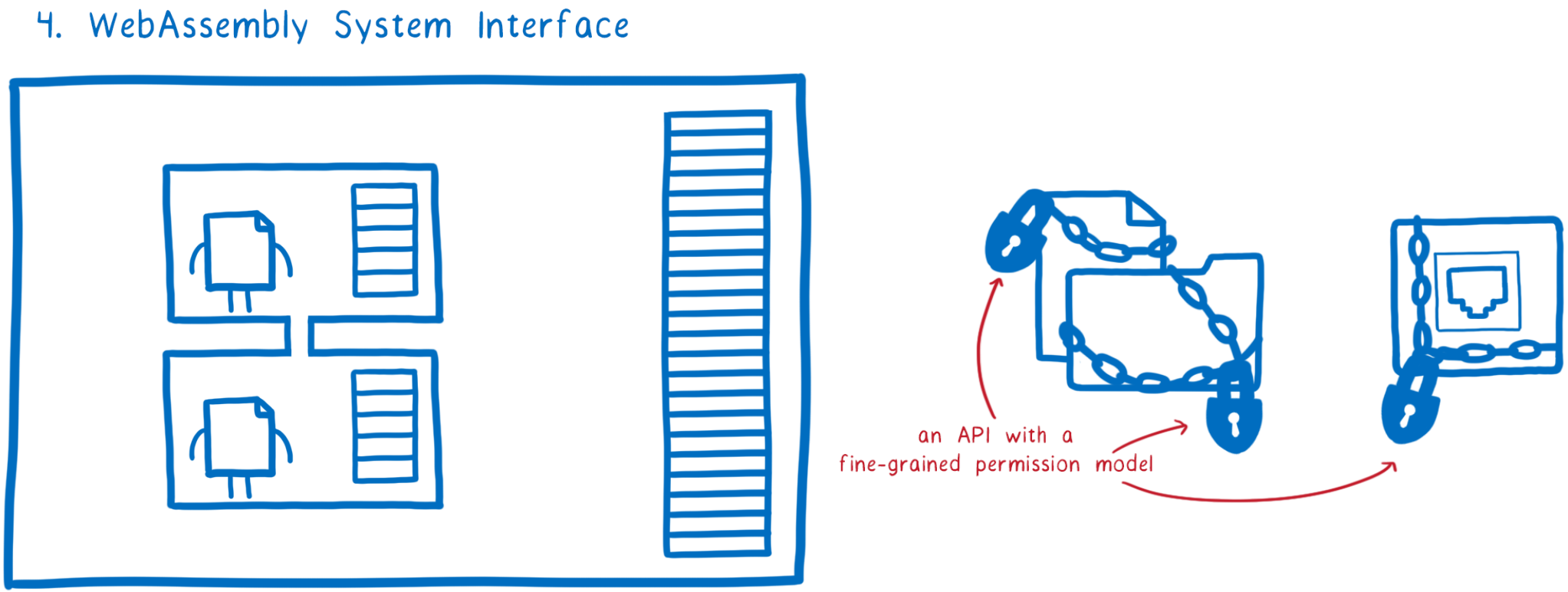


@nielstanis@infosec.exchange

WebAssembly Nano-Process

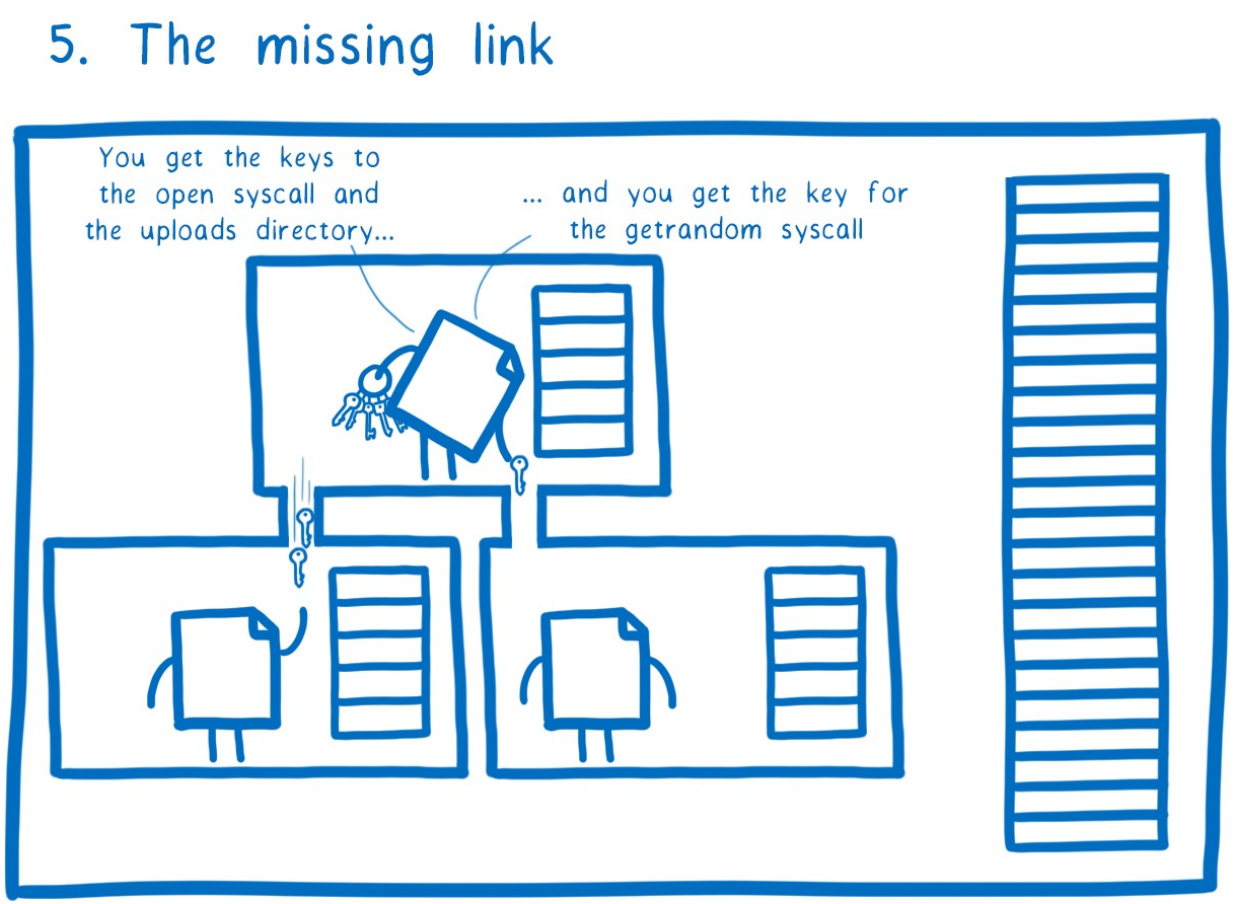


WebAssembly Nano-Process



@nielstanis@infosec.exchange

WebAssembly Nano-Process

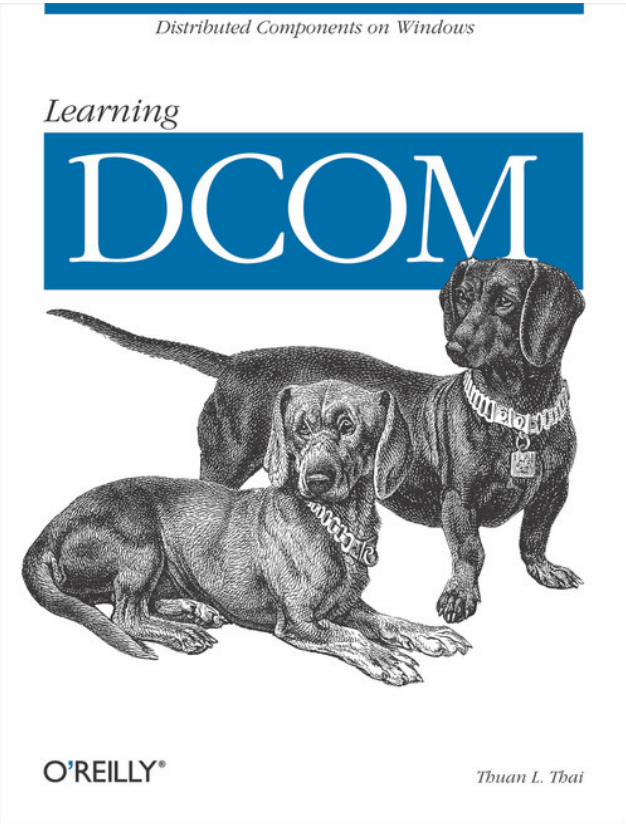


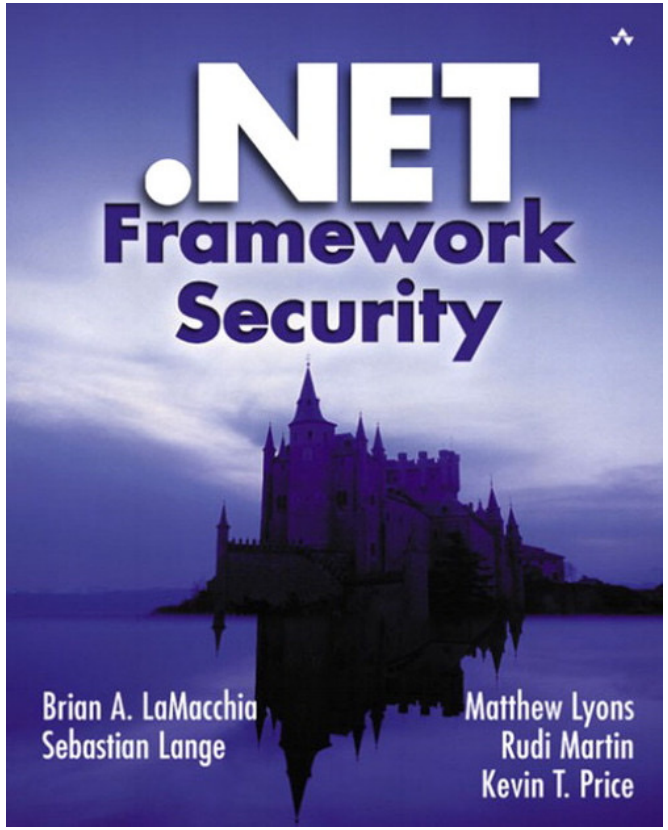
@nielstanis@infosec.exchange

WebAssembly Component Model

- Cloud Native WASM Day EU 2023
  - Evolution of Wasm: Past, Present, Future - Bailey Hayes, Cosmonic
    https://youtu.be/6_BRLqxiZPU
  - WASI and the Cloud - Jiaxiao Zhou, Microsoft & Dan Gohman, Fastly
    https://youtu.be/5WQRT62V_VU
  - Future of Component Tooling - Peter Huene & Guy Bedford, Fastly
    https://youtu.be/JCIwpc7x4jU

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

Conclusion

- Cloud Native ❤️ WebAssembly
- WebAssembly has a lot of potential to be used to run, extend, and secure your .NET applications
- Its as secure as the WebAssembly runtime implementation!
- I like top-down approach Bytecode Alliance is taking in moving forward, feedback will change/influence

Questions?

- https://github.com/nielstanis/codeeurope2023-wasm
- ntanis at Veracode.com
- @nielstanis@infosec.exchange
- https://blog.fennec.dev
- Dziękuję! Thank you!