OWASP
BeNeLux Days
Conference
2023

Using WebAssembly to run, extend, and secure your application

Niels Tanis

VERACODE

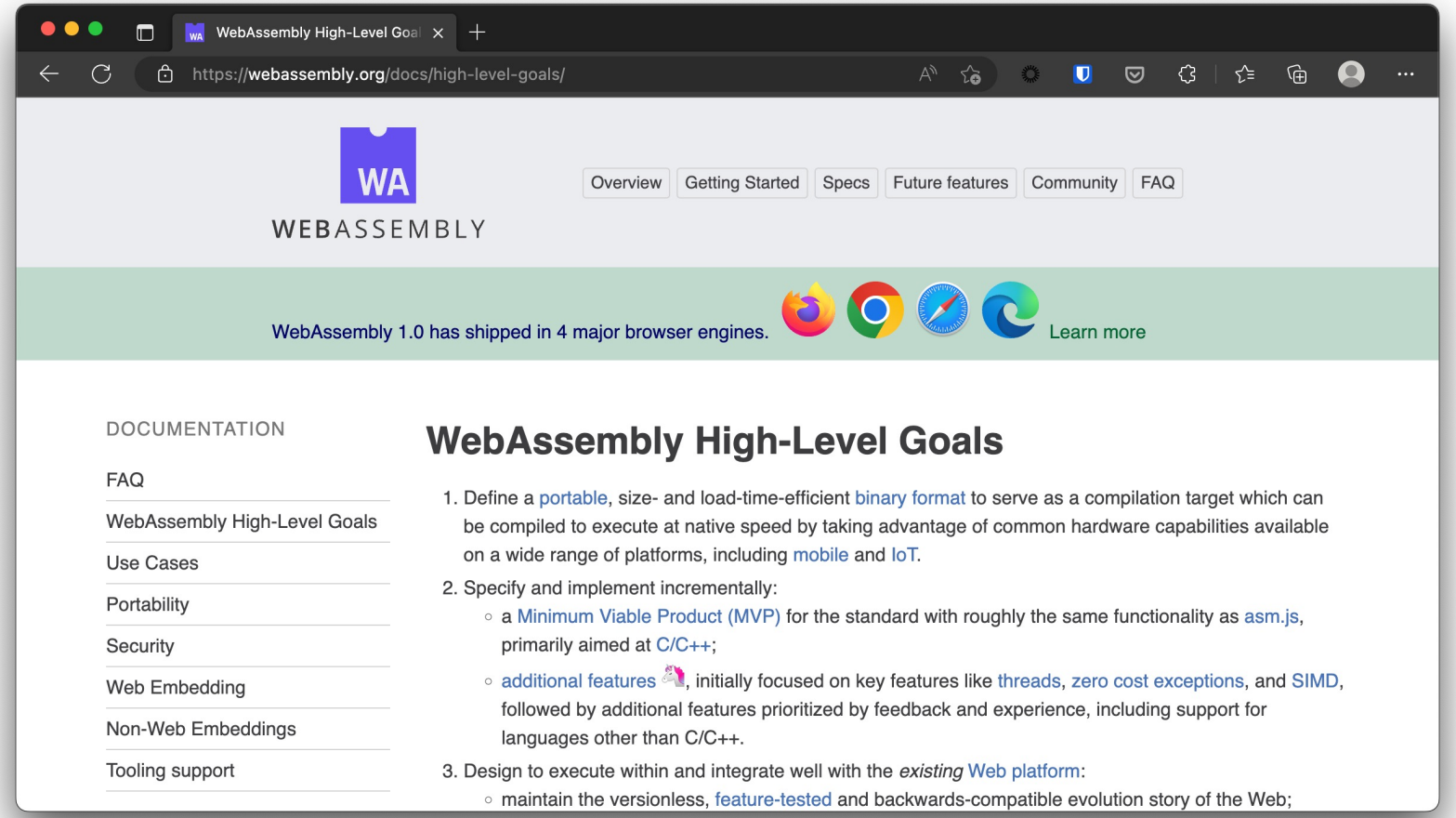
Who am I?

- 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

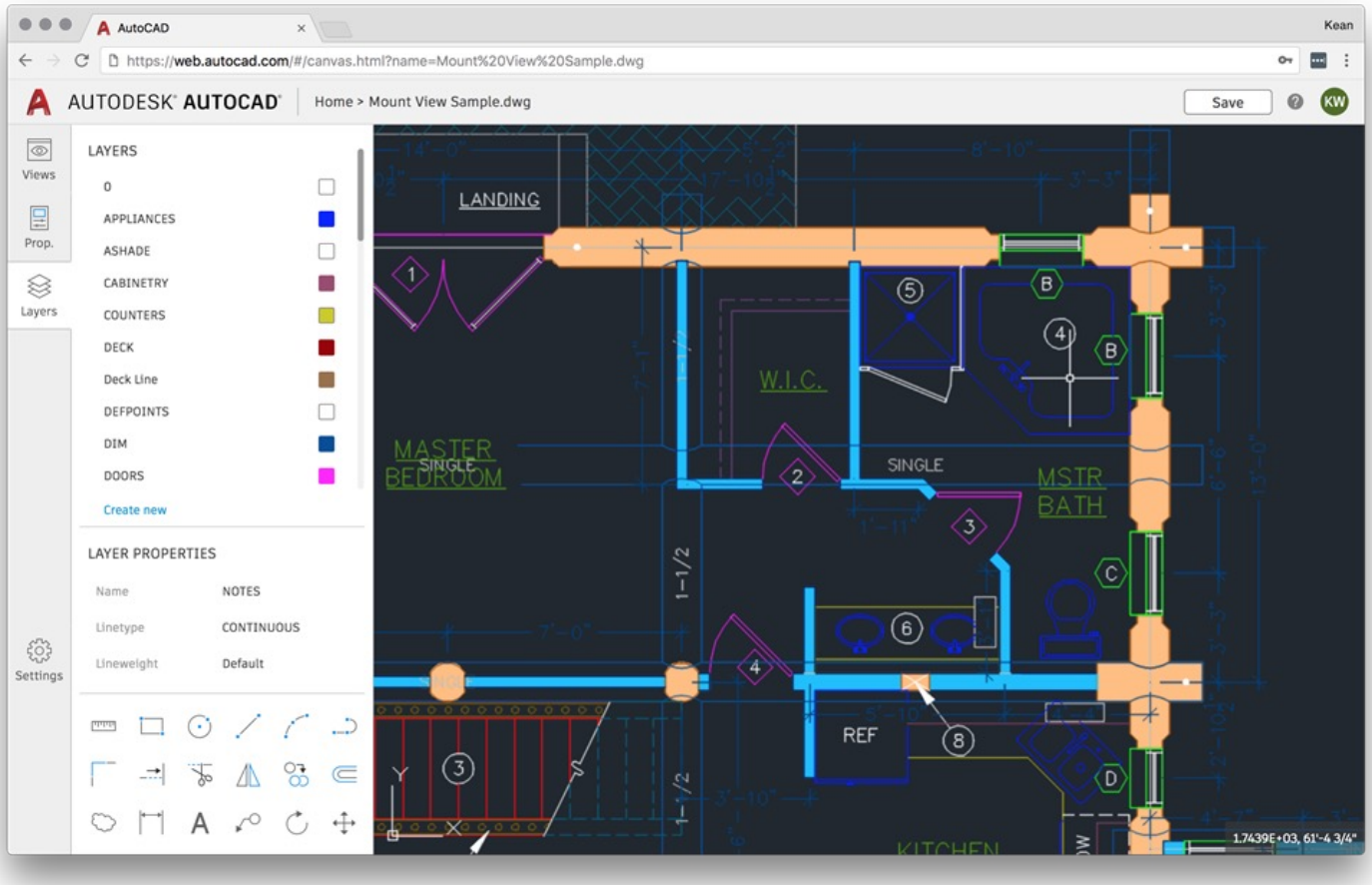
VERACODE

MVP Microsoft® Most Valuable Professional

WebAssembly

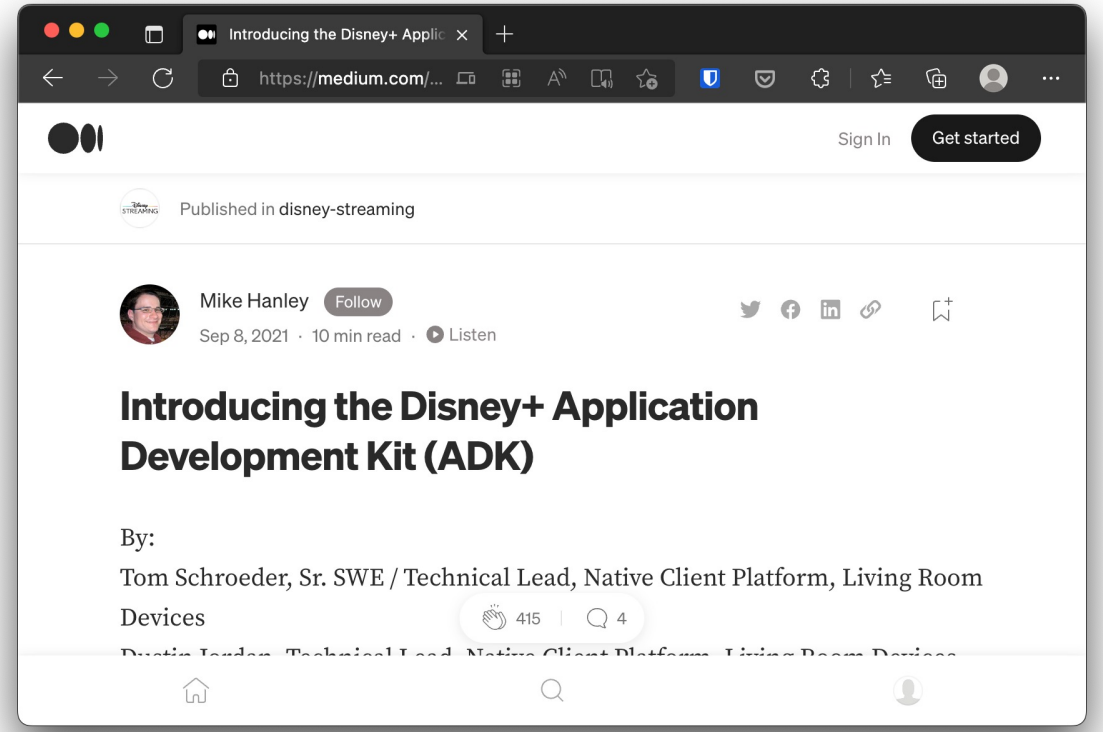


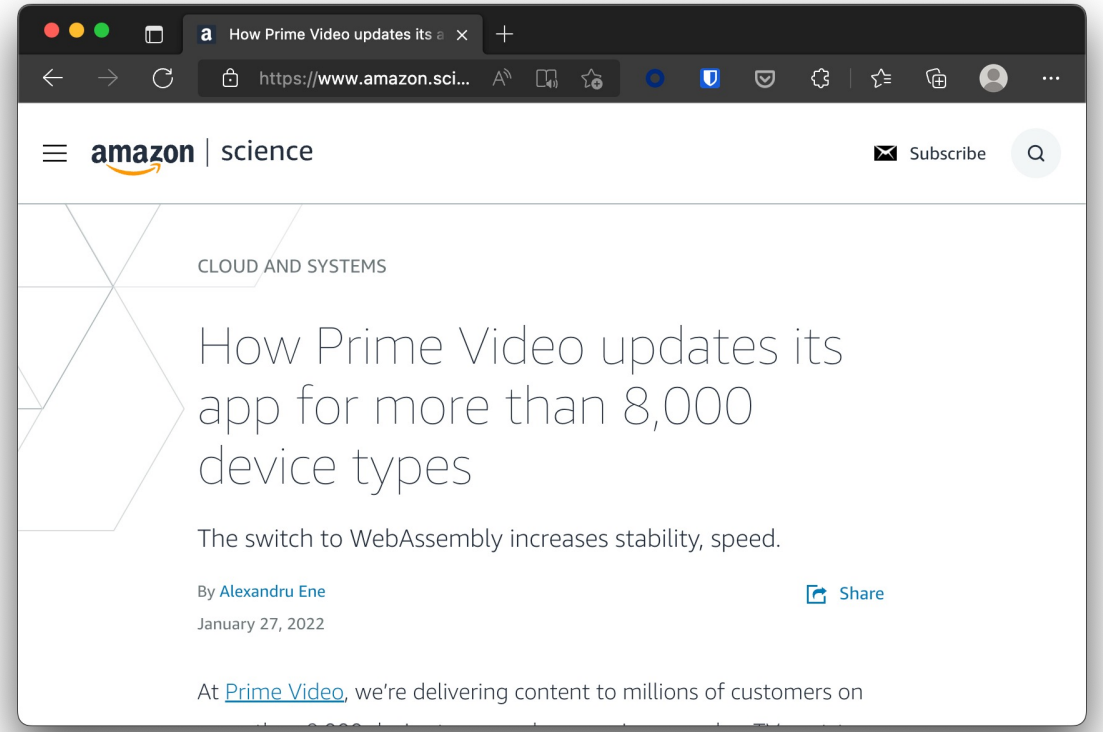
WebAssembly - AutoCAD



@nielstanis@infosec.exchange

WebAssembly - SDK's





@nielstanis@infosec.exchange

Agenda

- Introduction
- WebAssembly 101
- Using WebAssembly to …
  - … run your application
  - … extend your application
  - .. secure your application
- 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

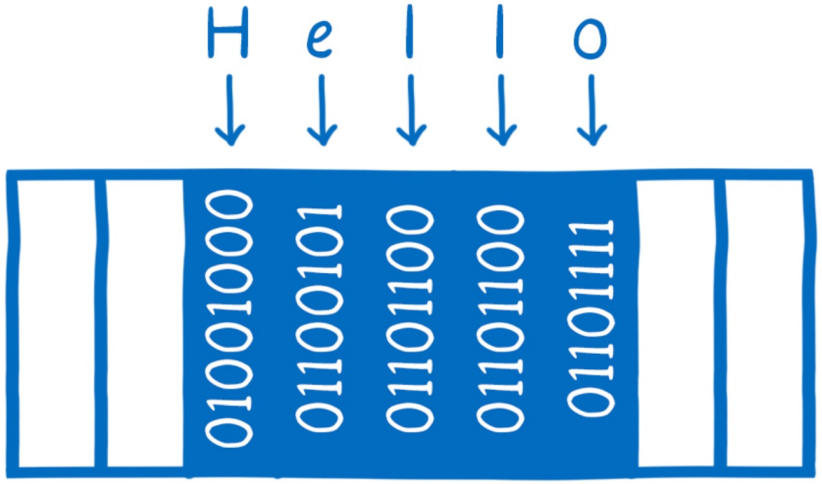
WEBASSEMBLY

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
- 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



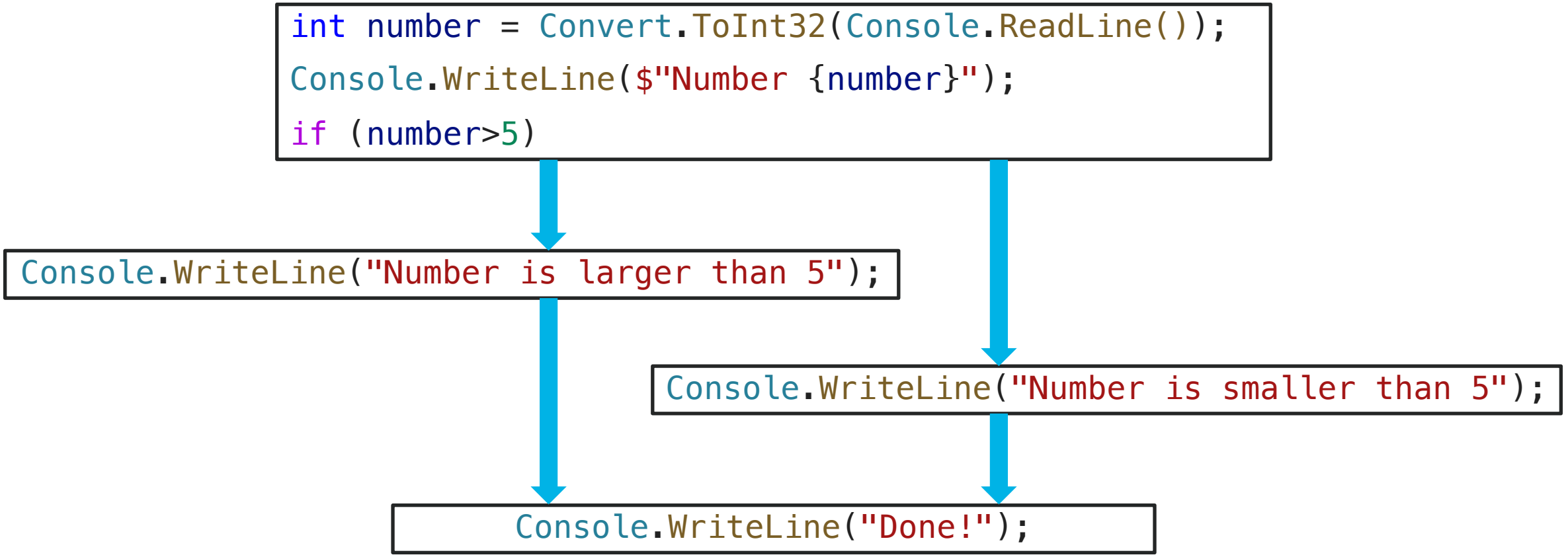
@nielstanis@infosec.exchange

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!");
```

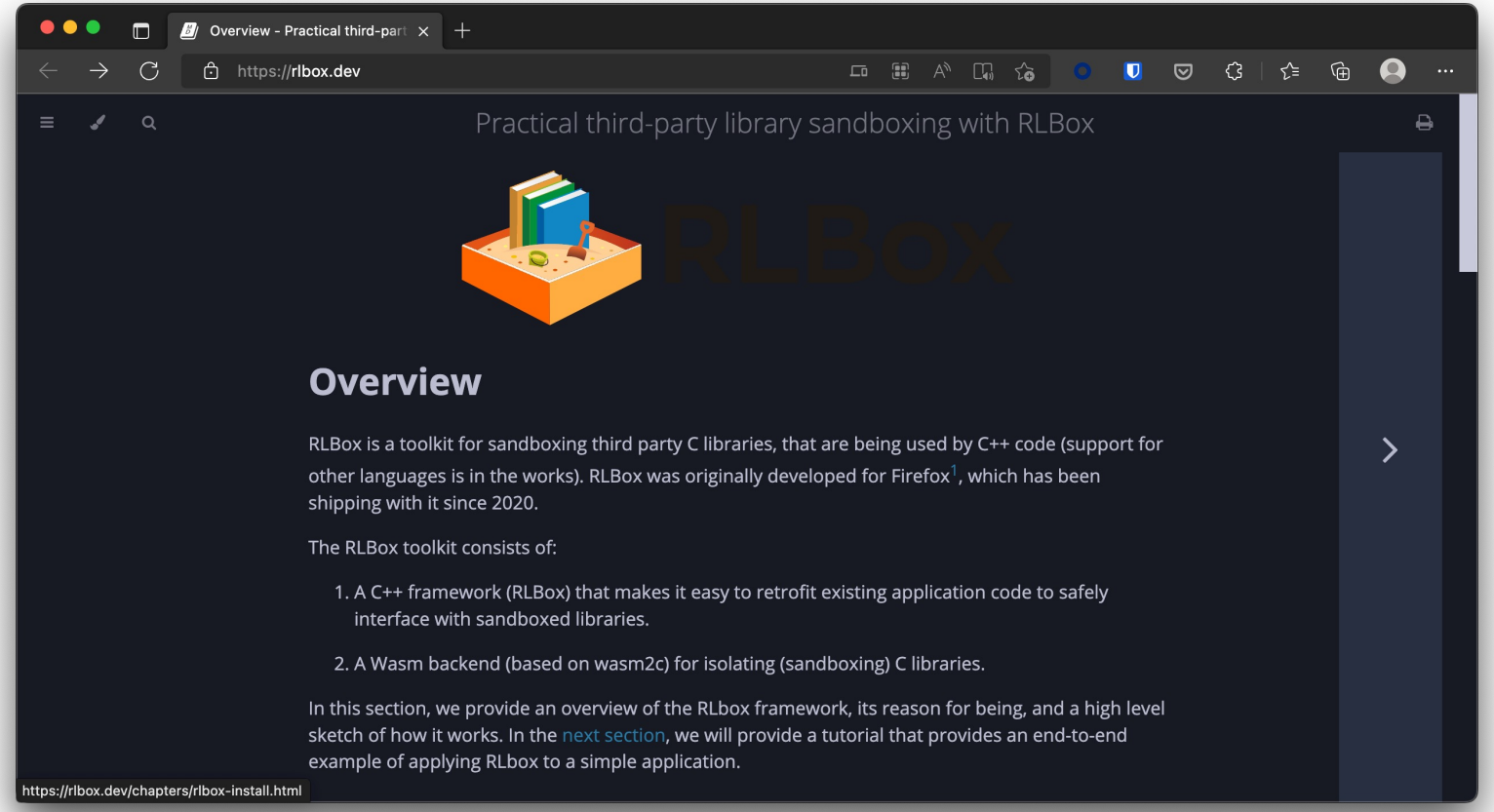

@nielstanis@infosec.exchange

WebAssembly Control-Flow Integrity



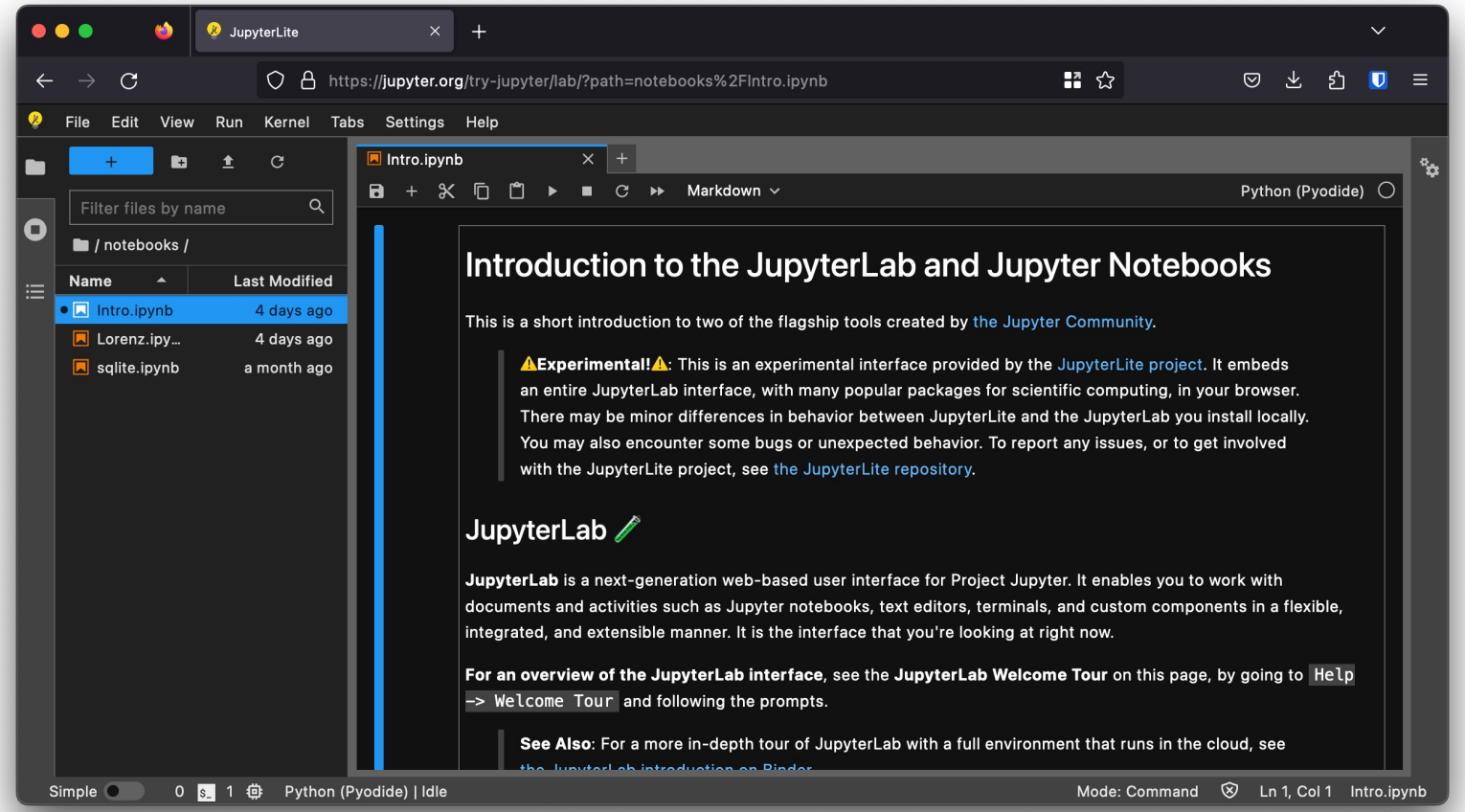
@nielstanis@infosec.exchange

FireFox RLBox



@nielstanis@infosec.exchange

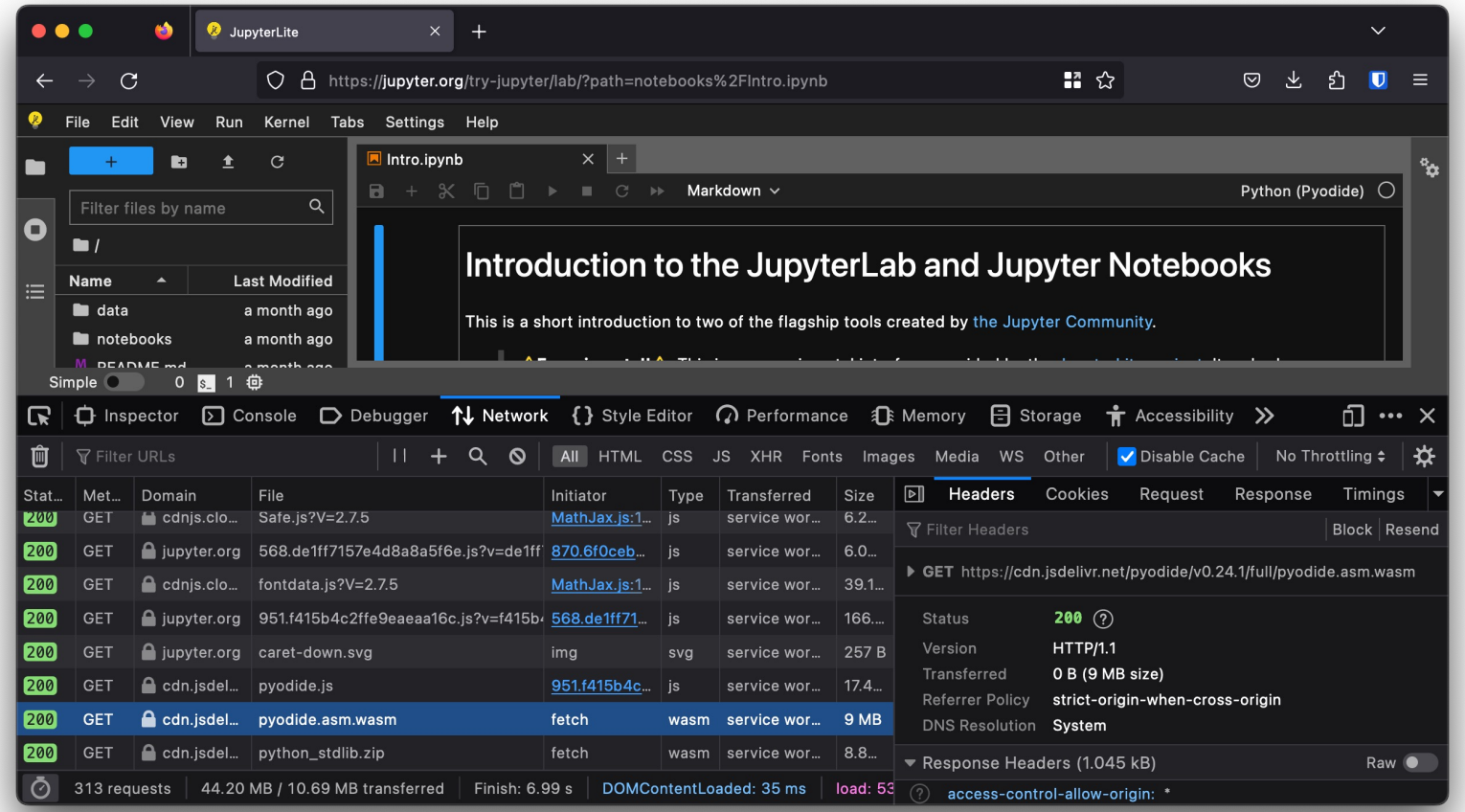
Python on WASM



OWASP®

@nielstanis@infosec.exchange

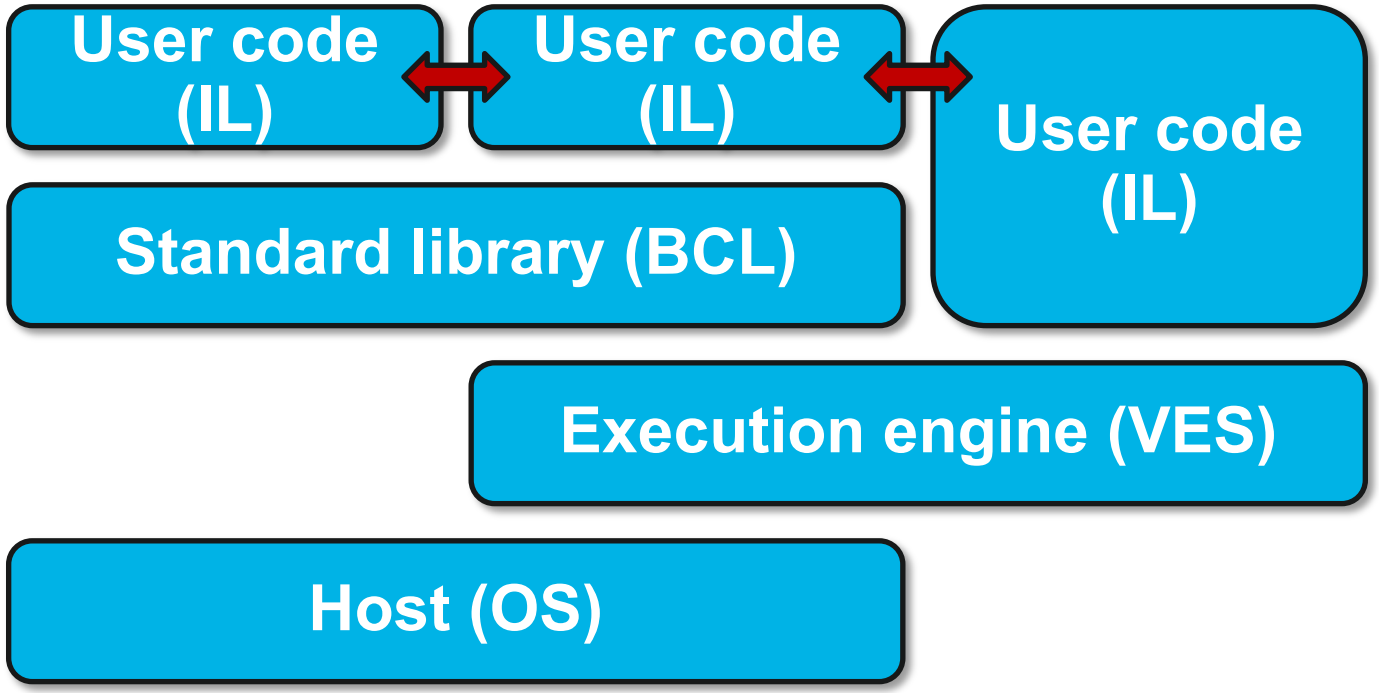
Python on WASM



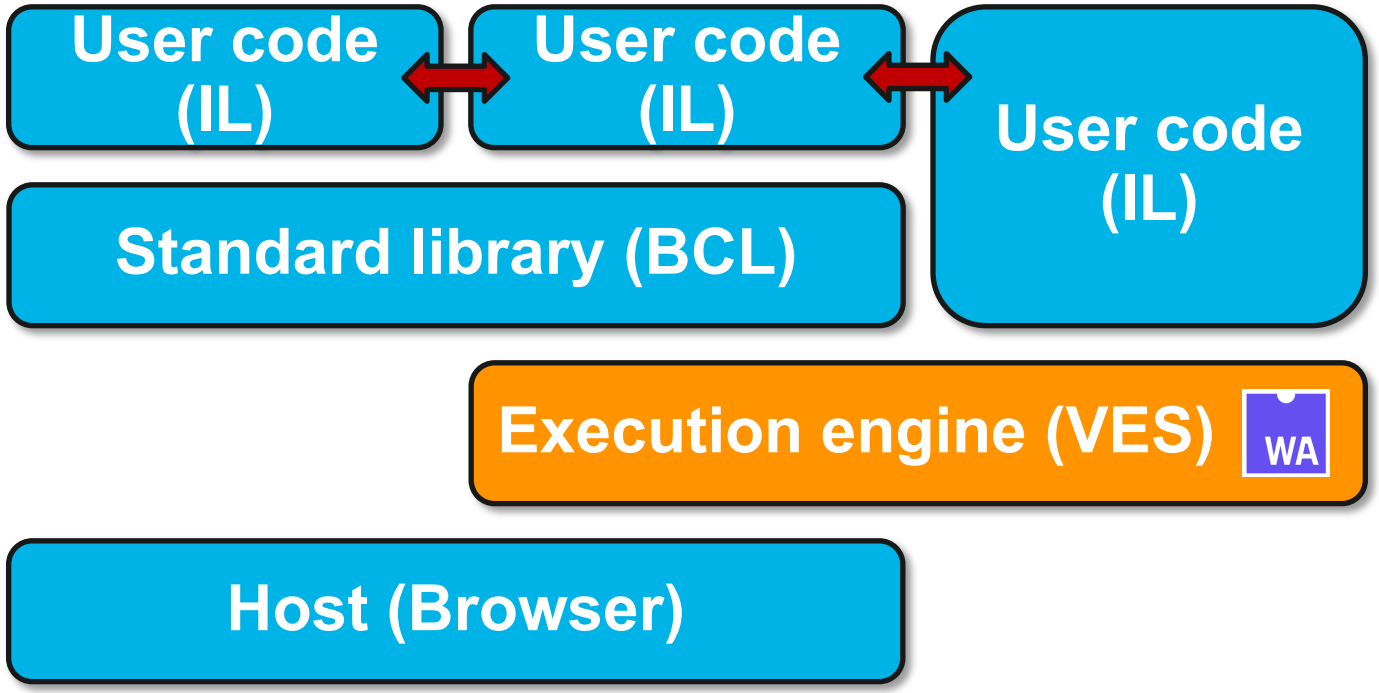
OWASP®

@nielstanis@infosec.exchange

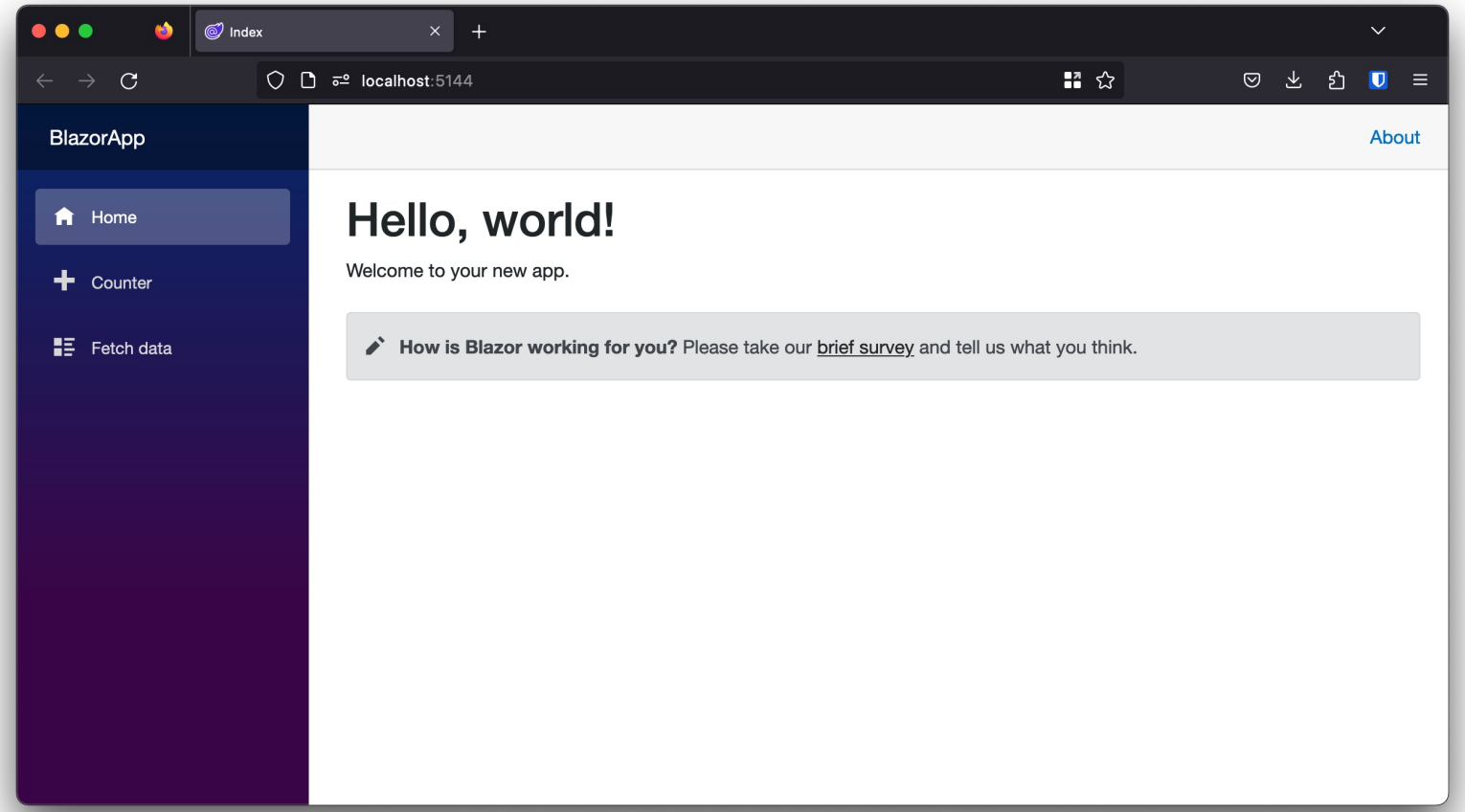
Running .NET on WebAssembly



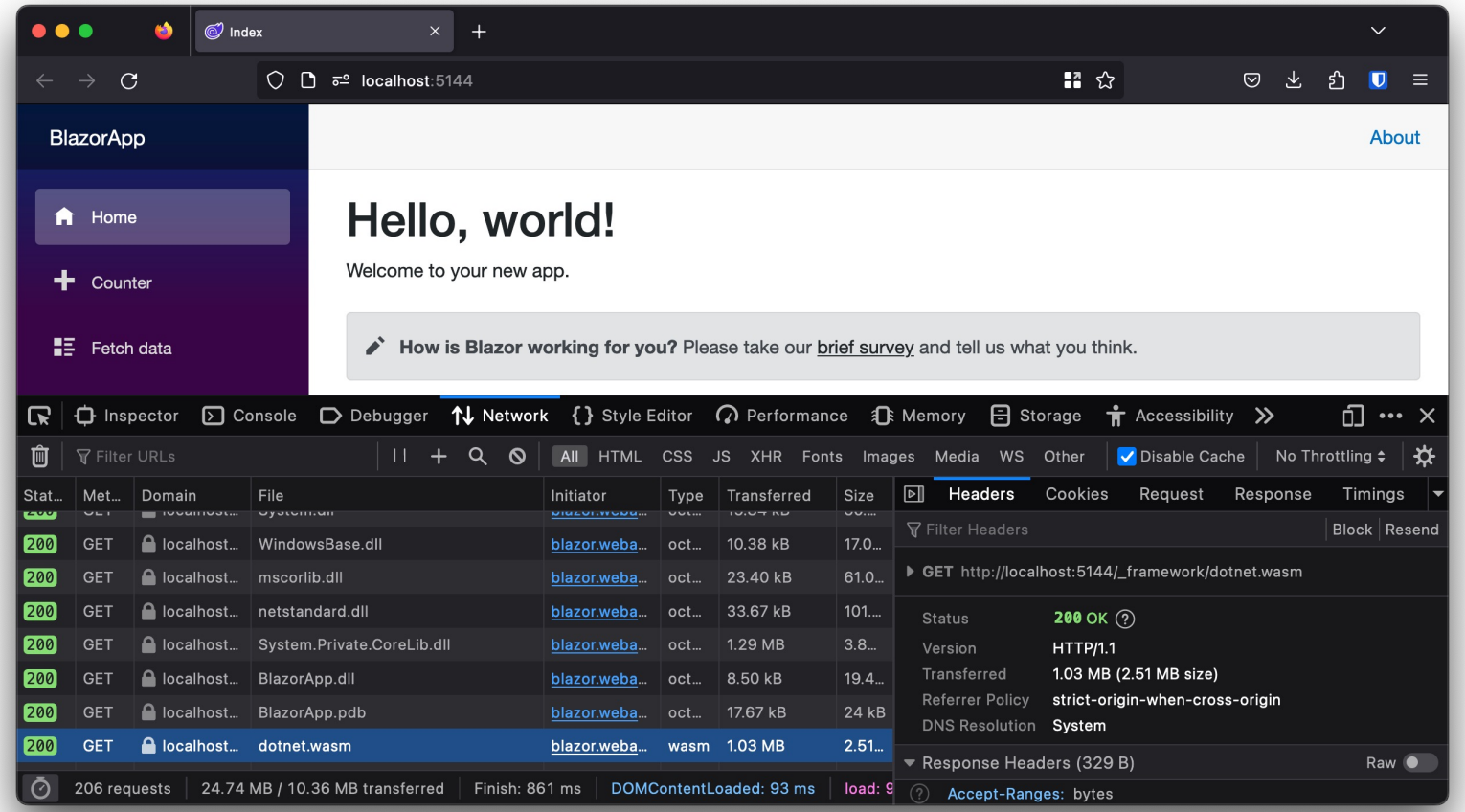
Running .NET on WebAssembly



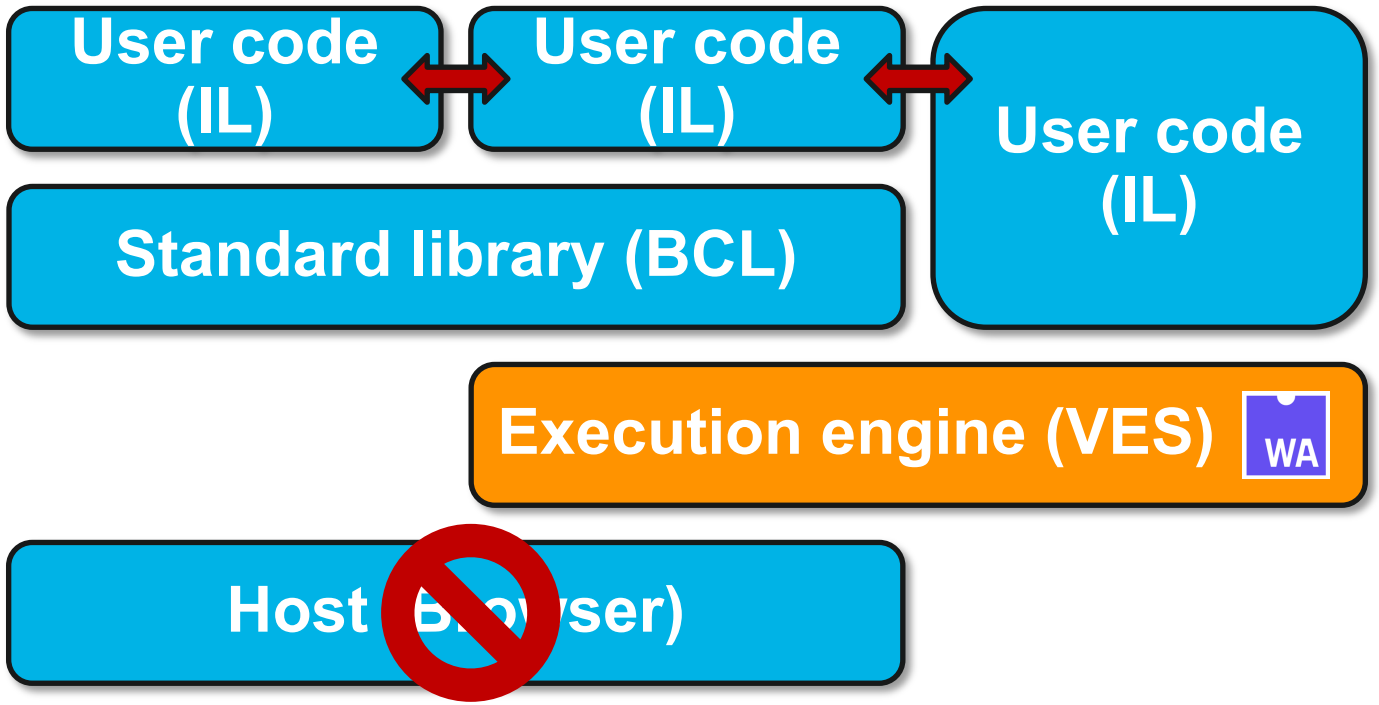
Blazor WebAssembly



Blazor WebAssembly



Running .NET on WebAssembly



@nielstanis@infosec.exchange

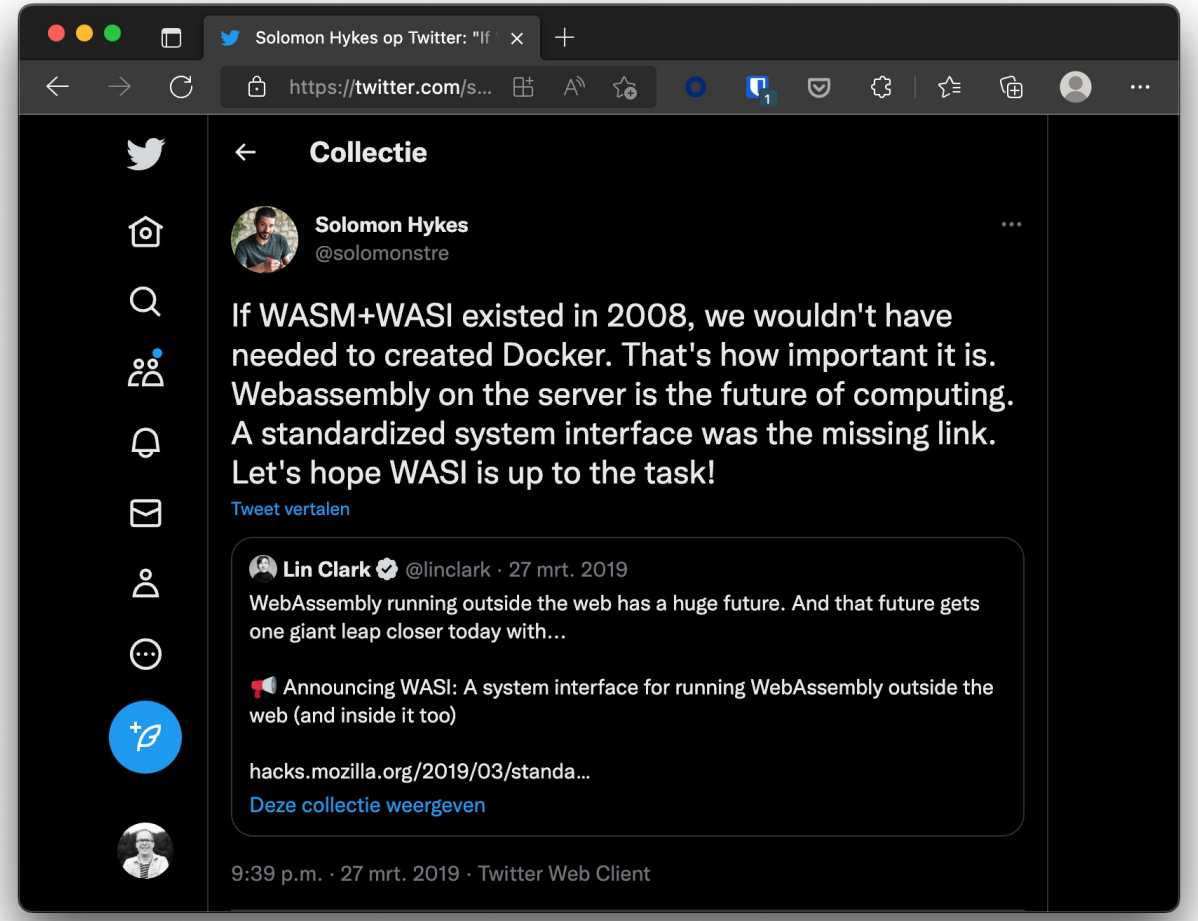
WebAssembly System Interface WASI

- 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 familiar with .NET Standard? ☺

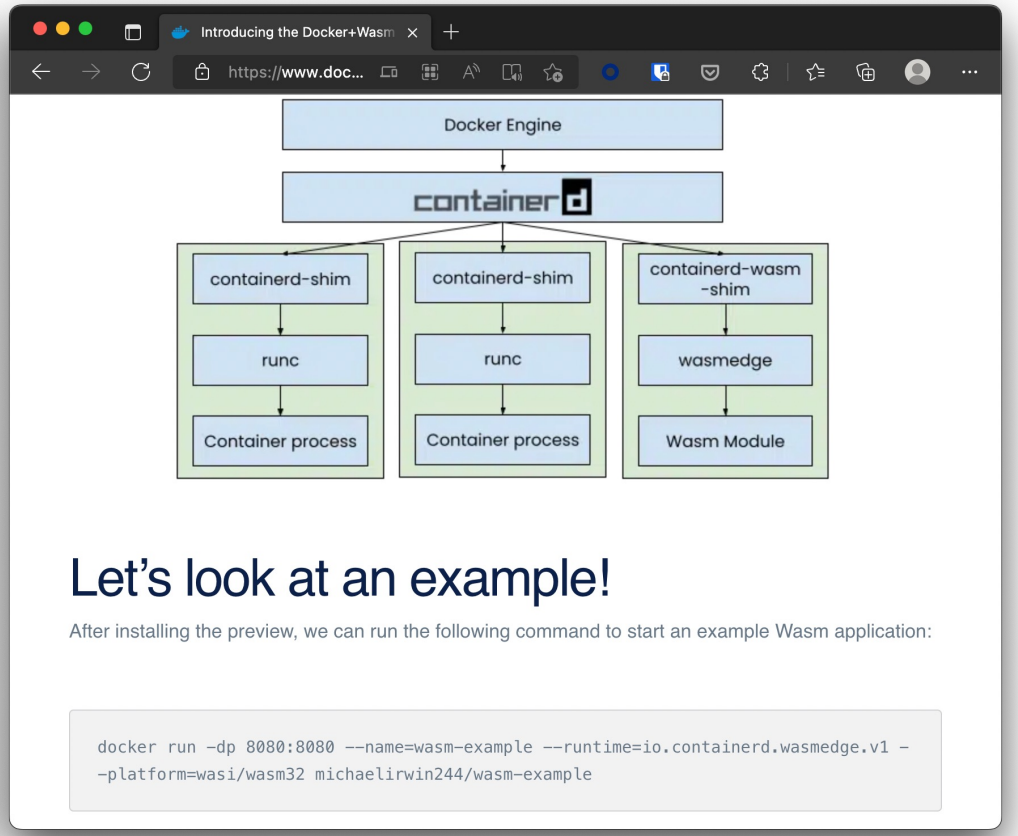
Docker vs WASM & WASI



Docker vs WASM & WASI

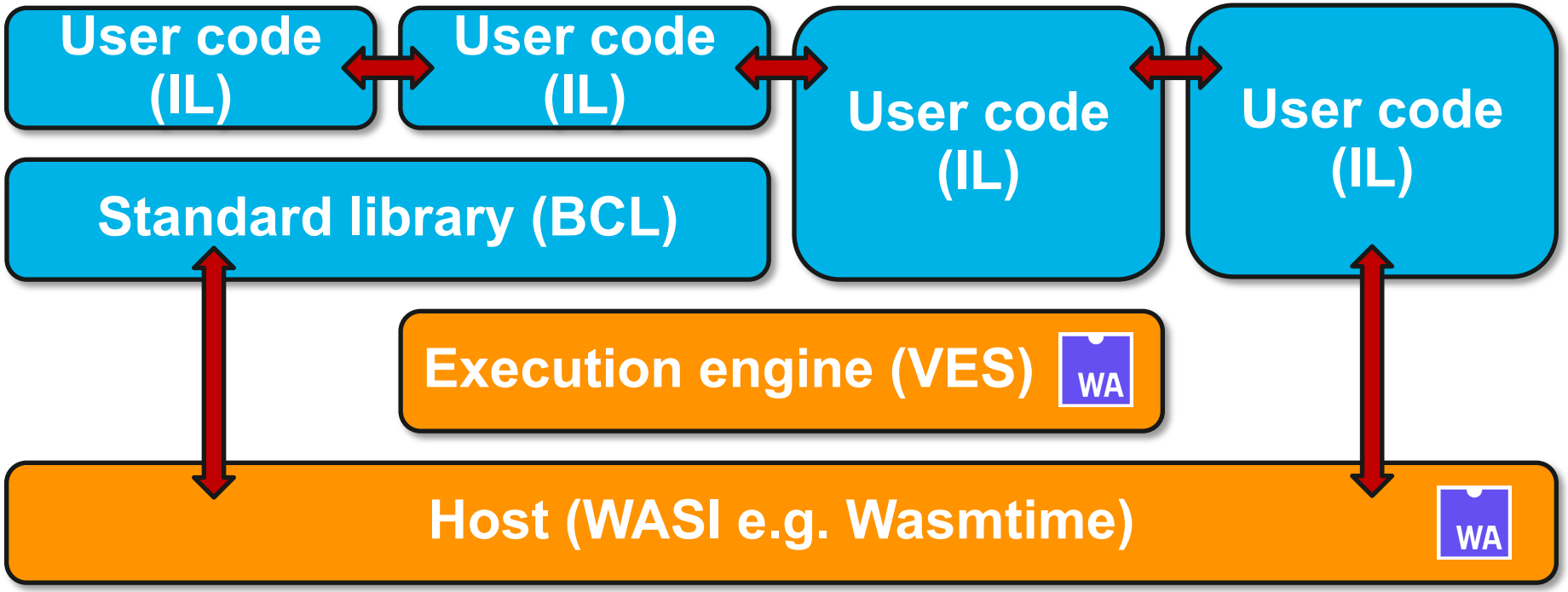
@nielstanis@infosec.exchange

Docker & WASM



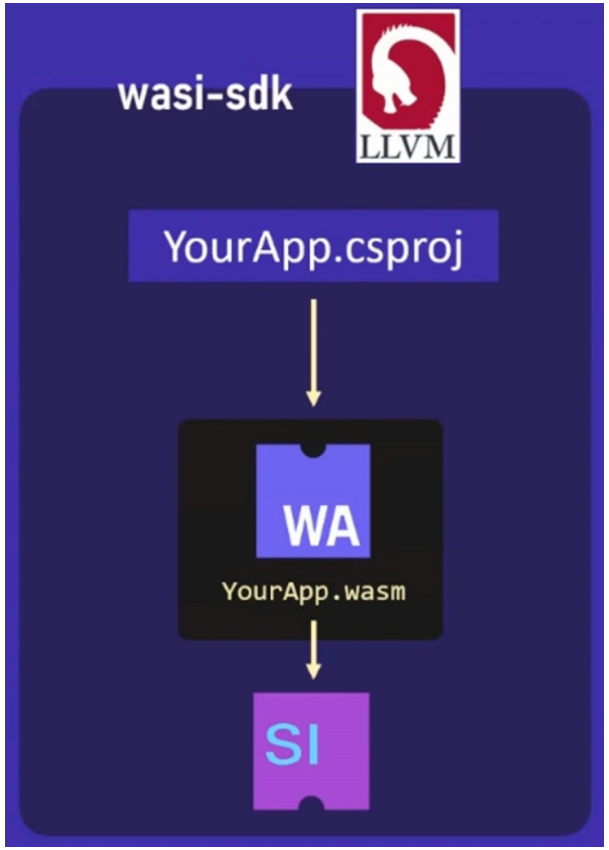
@nielstanis@infosec.exchange

WebAssembly System Interface WASI



@nielstanis@infosec.exchange

Experimental WASI SDK for .NET

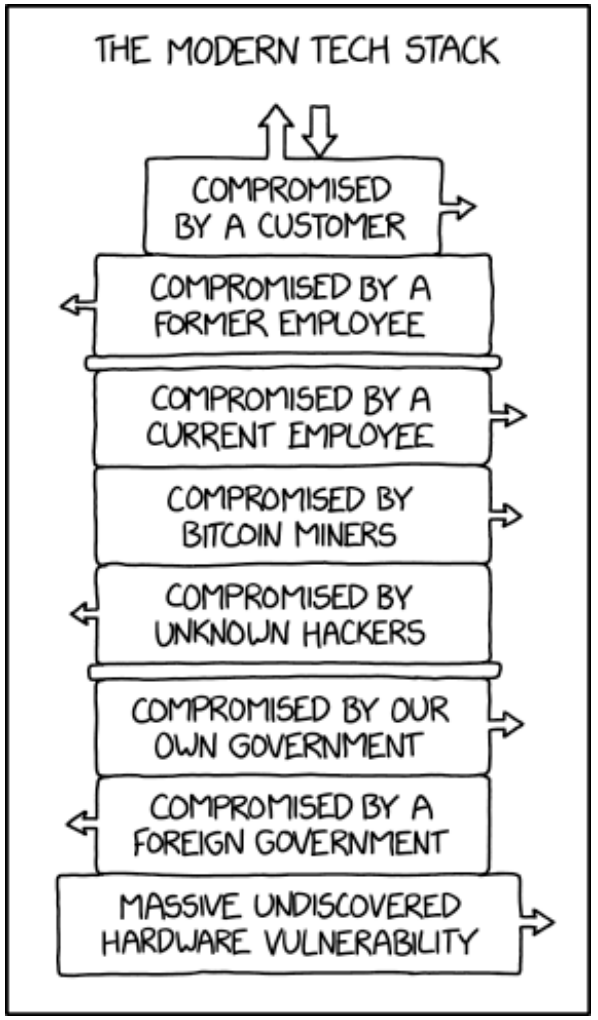


@nielstanis@infosec.exchange

Extending .NET with WASM

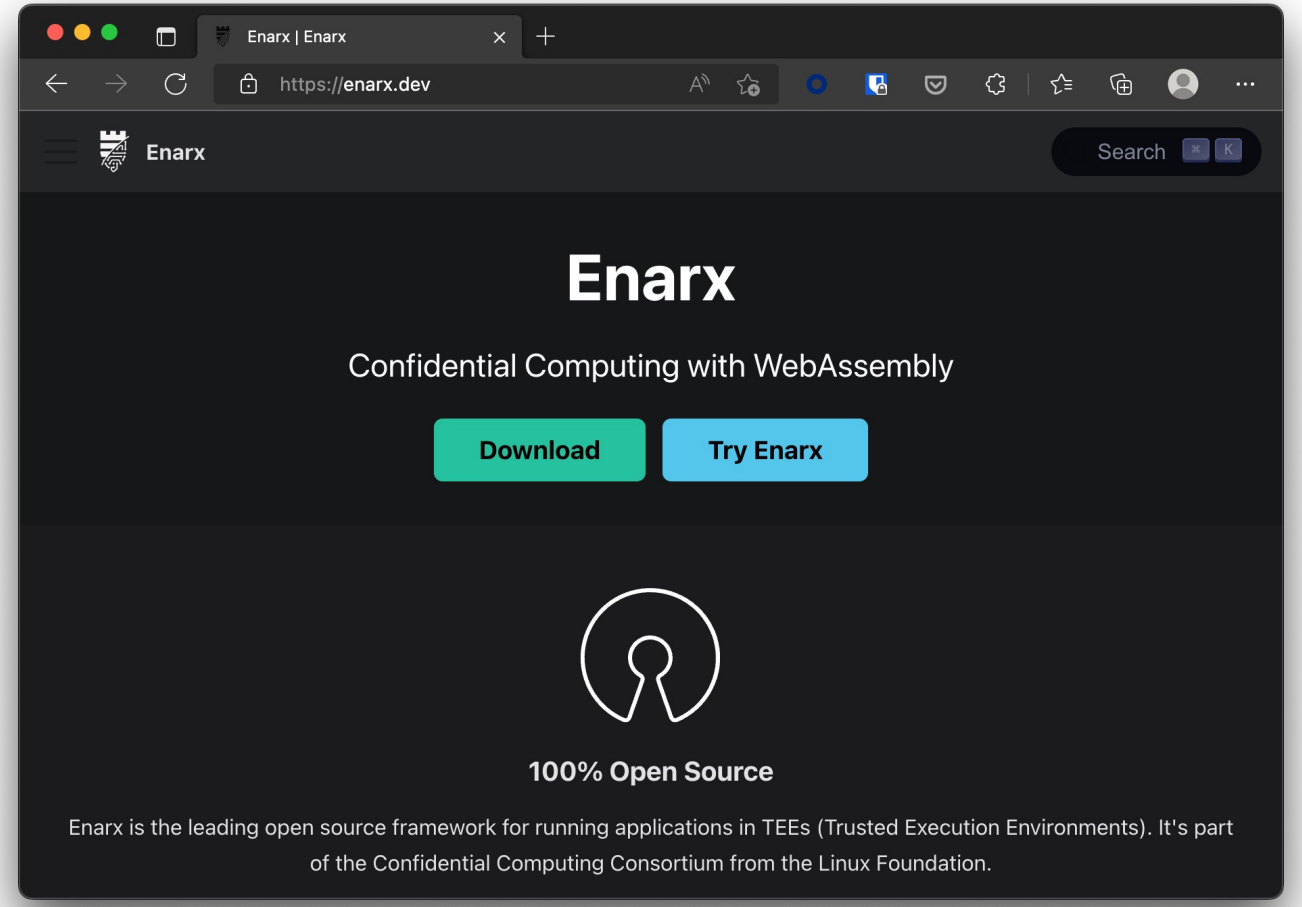
- WasmTime.NET NuGet package
- Can run WASM inside of any .NET application
- Extend with Rust based WASM module
- Limit capabilities
- Demo time!

Trusted Computing – XKCD 2166



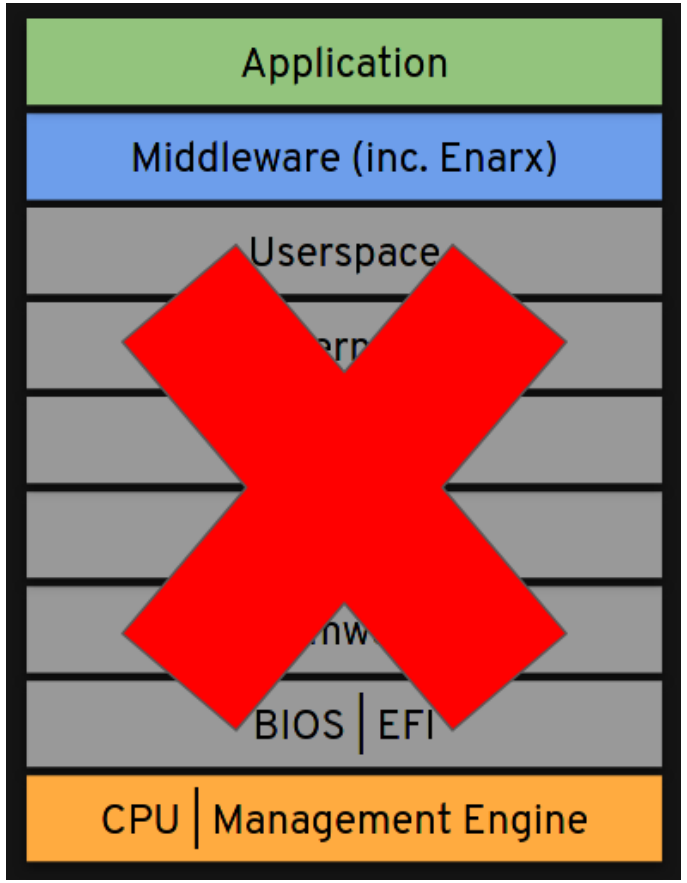
@nielstanis@infosec.exchange

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

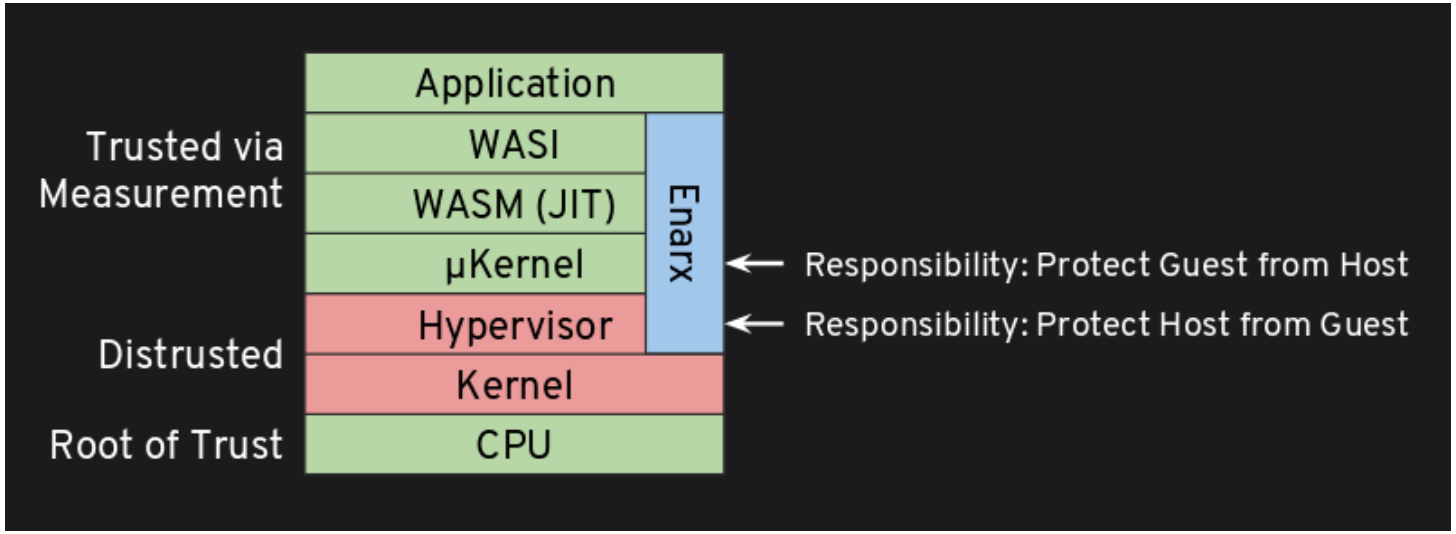
- Don’t trust the host
- Don’t trust the host owner
- Don’t trust the host operator
- Hardware cryptographically verified
- Software audited and cryptographically verified



@nielstanis@infosec.exchange

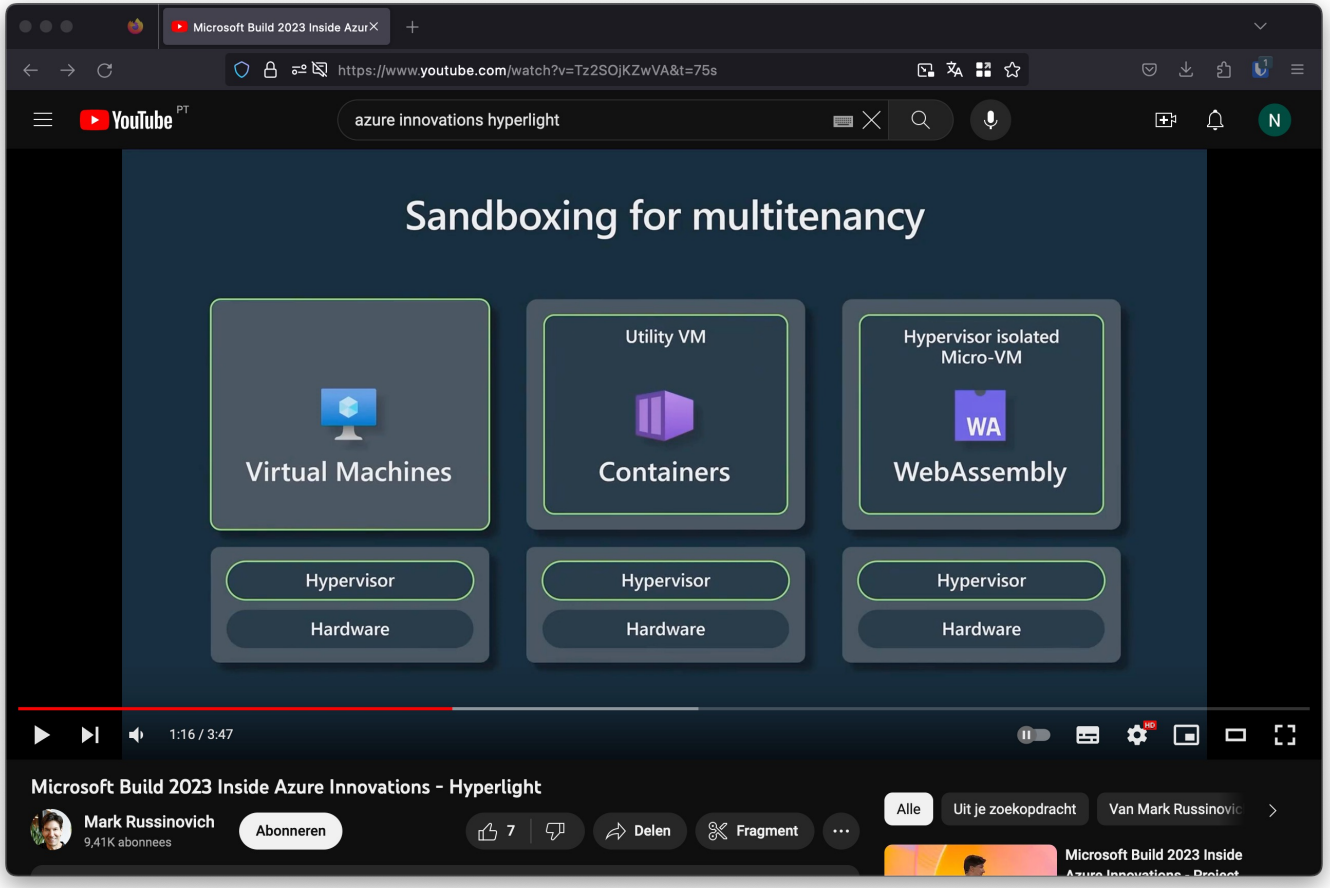
Enarx

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

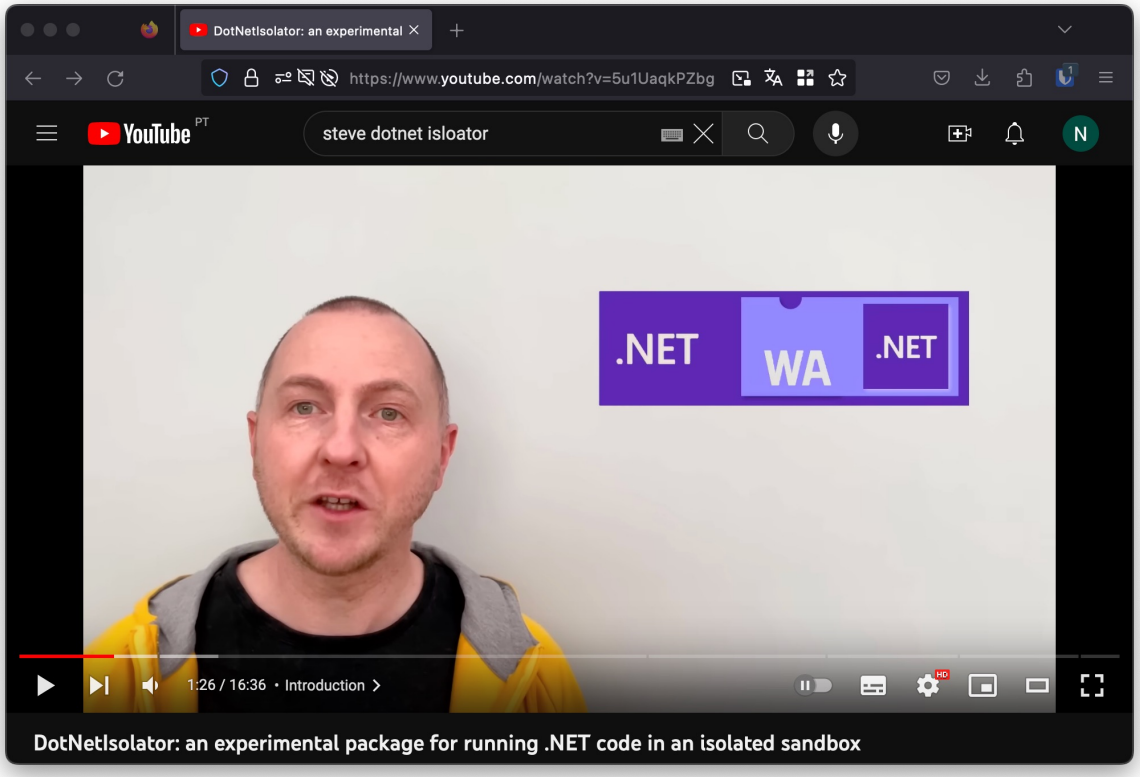


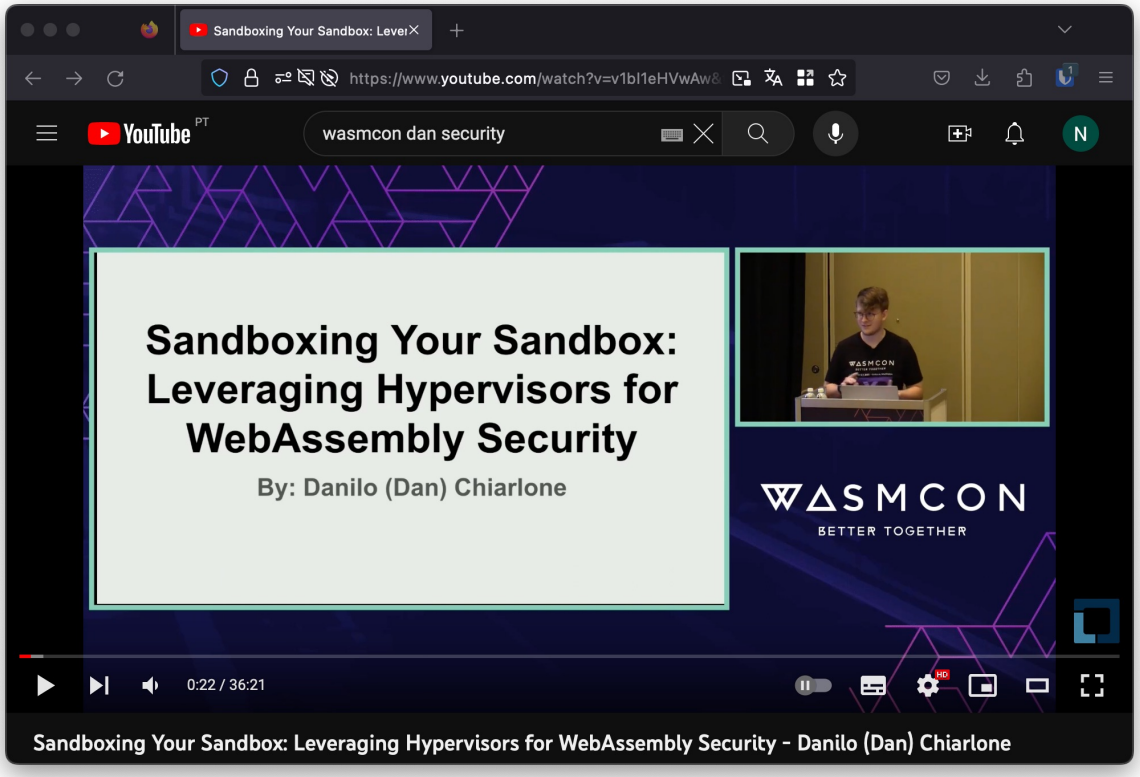
@nielstanis@infosec.exchange

Project Hyperlight

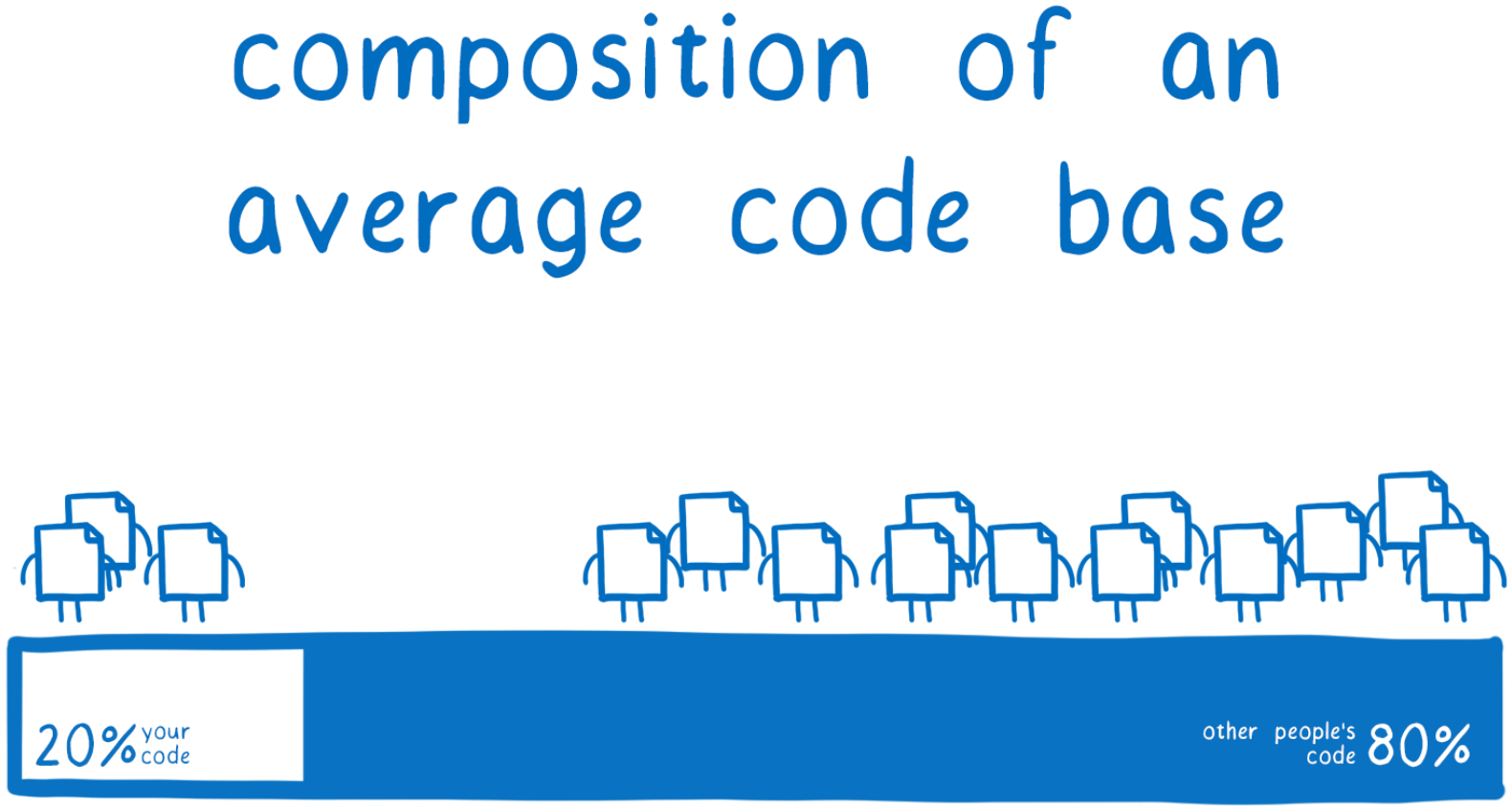


DotNetIsolator & Project Hyperlight



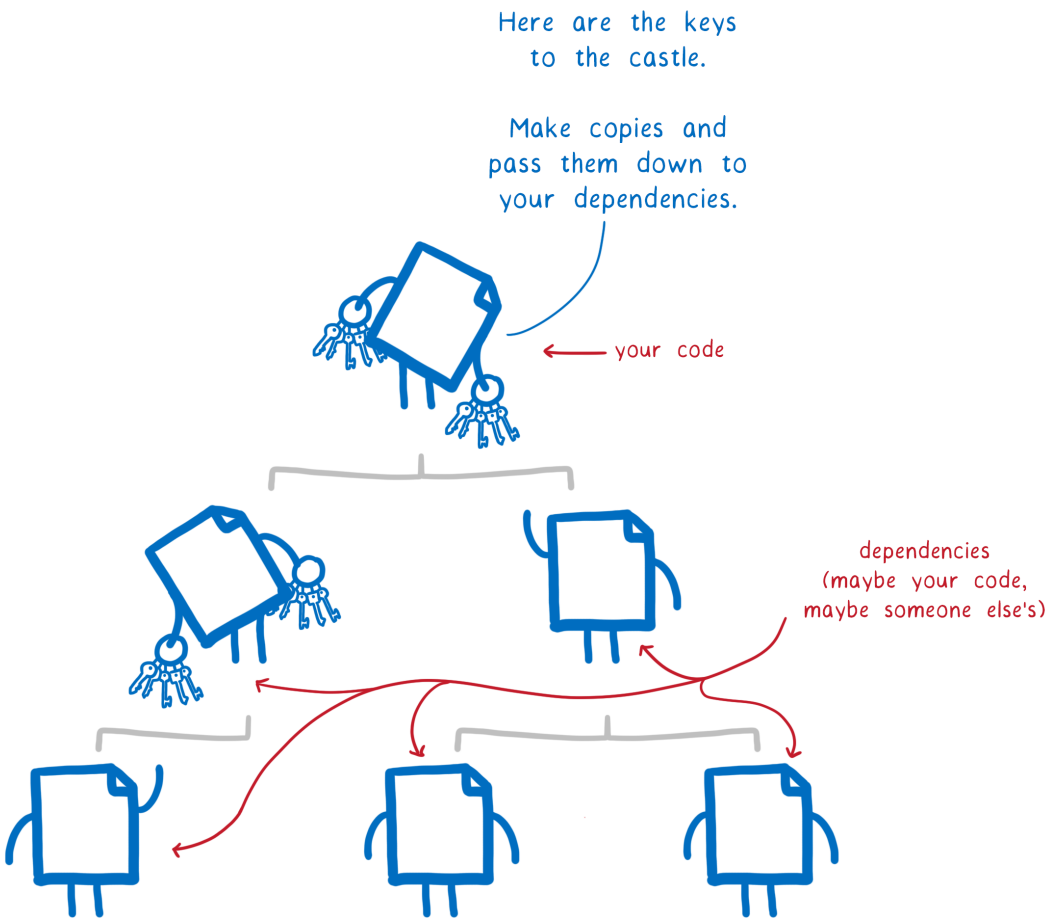


WASM – What's next?



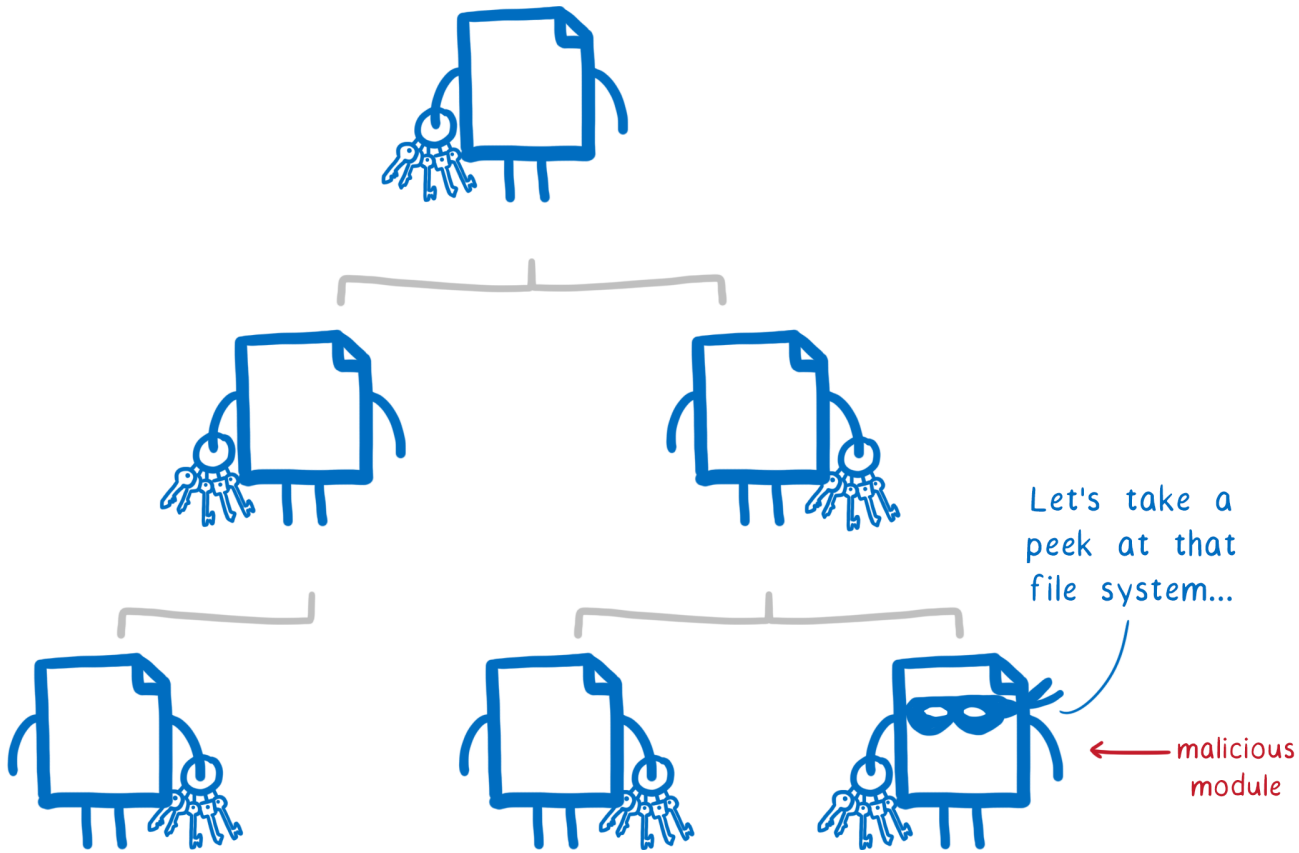
@nielstanis@infosec.exchange

Dependencies



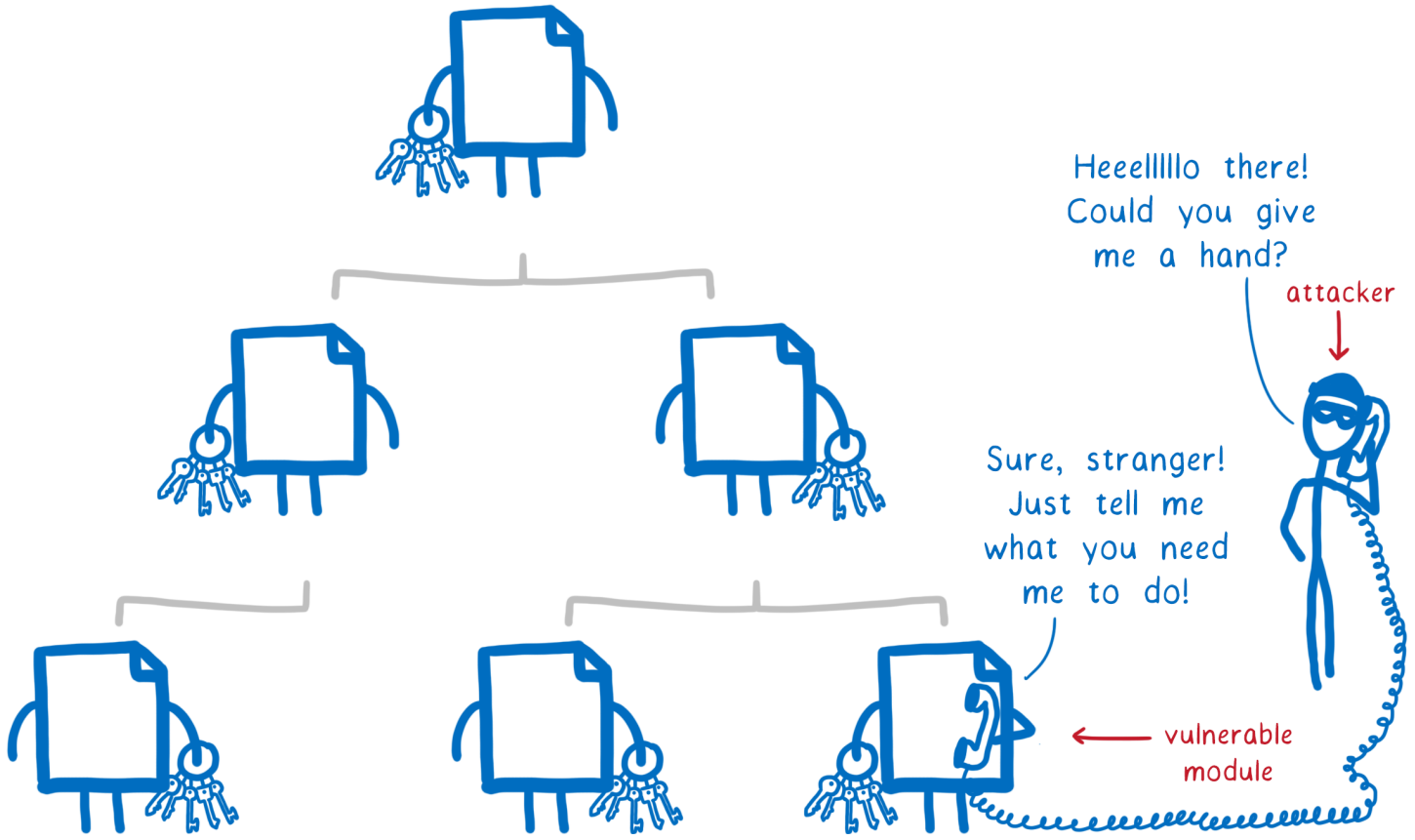
@nielstanis@infosec.exchange

Malicious module



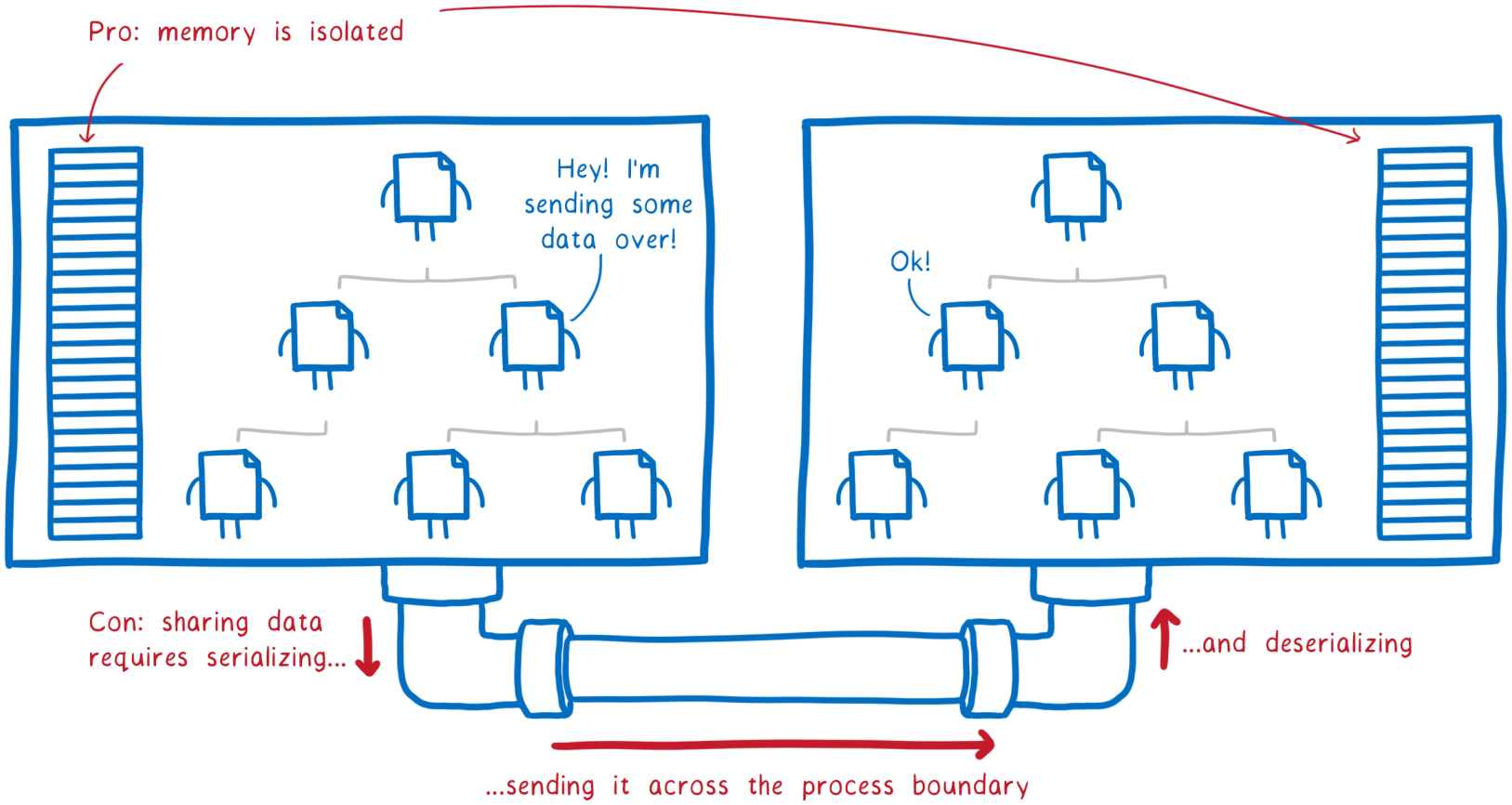
@nielstanis@infosec.exchange

Vulnerable module



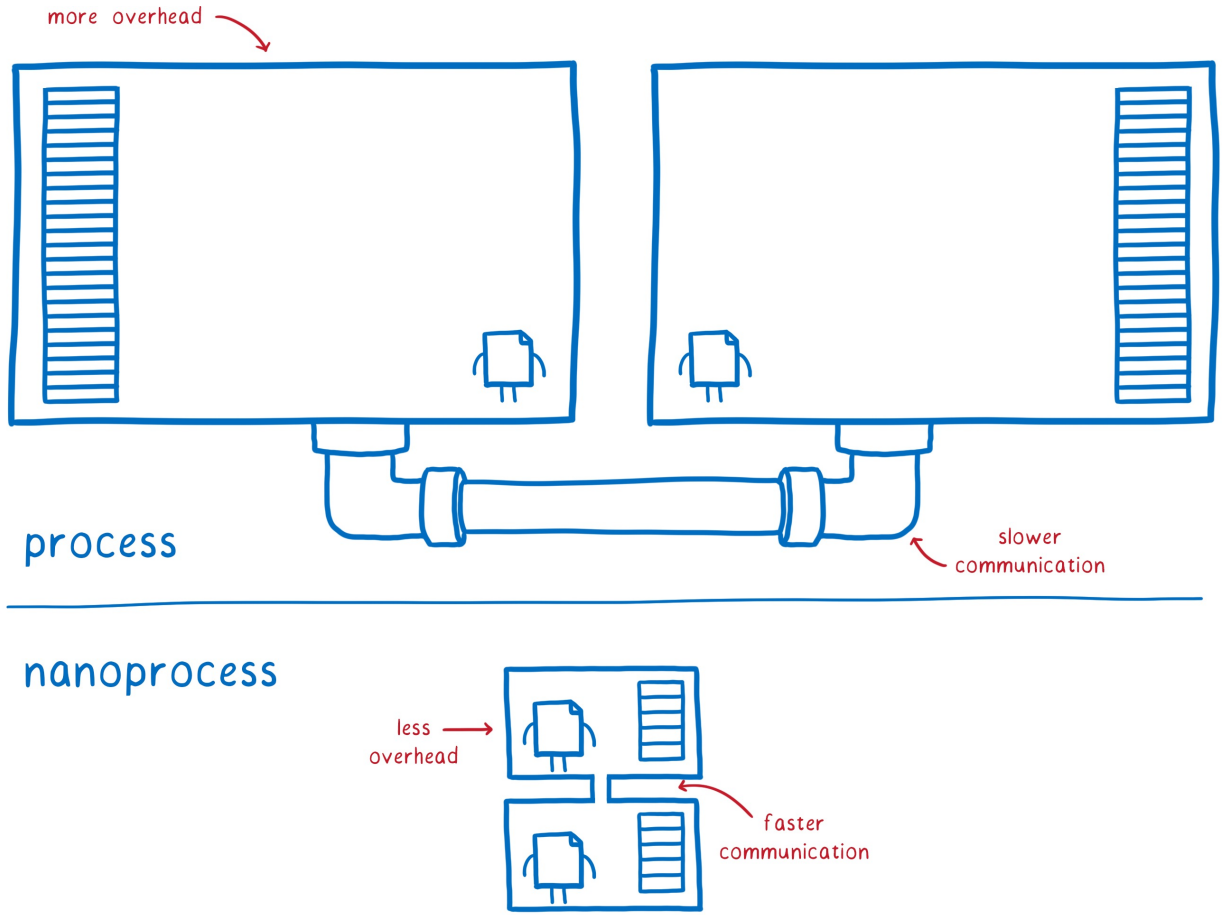
@nielstanis@infosec.exchange

Process Isolation



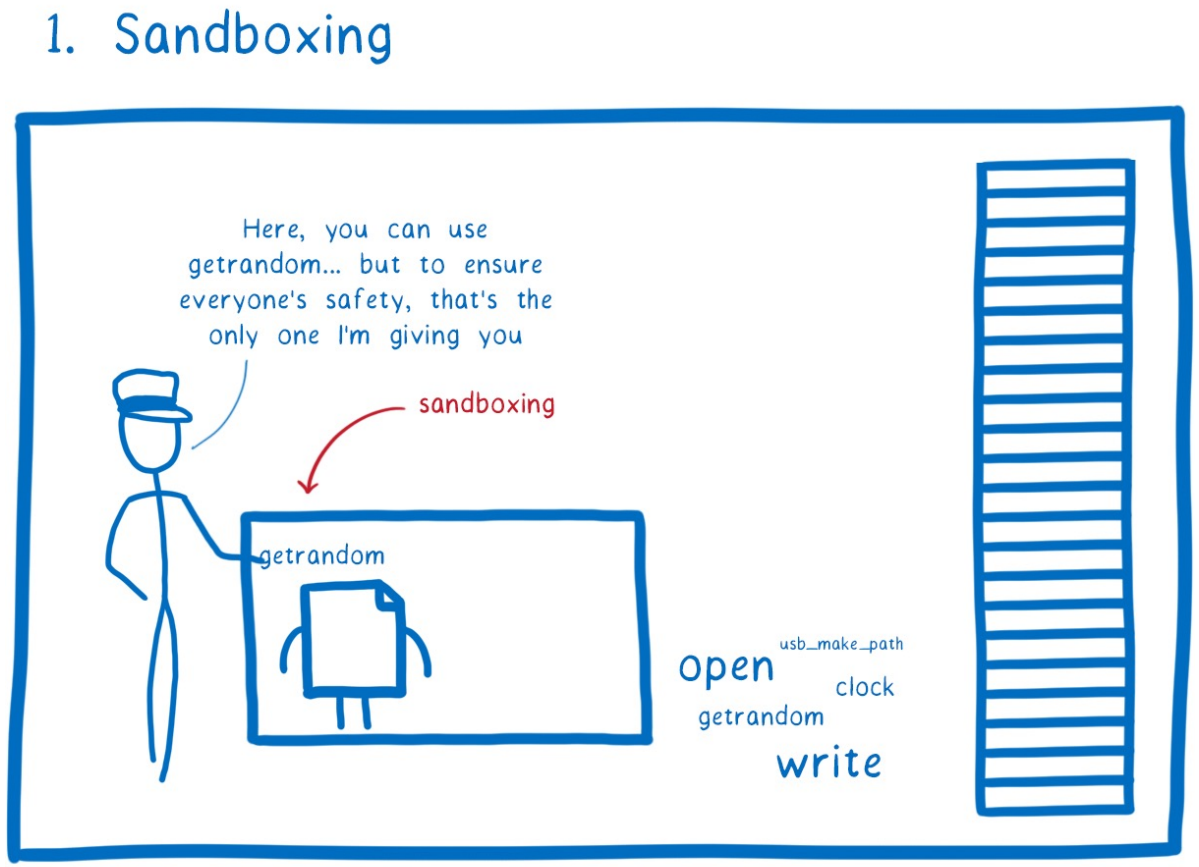
@nielstanis@infosec.exchange

WebAssembly Nano-Process



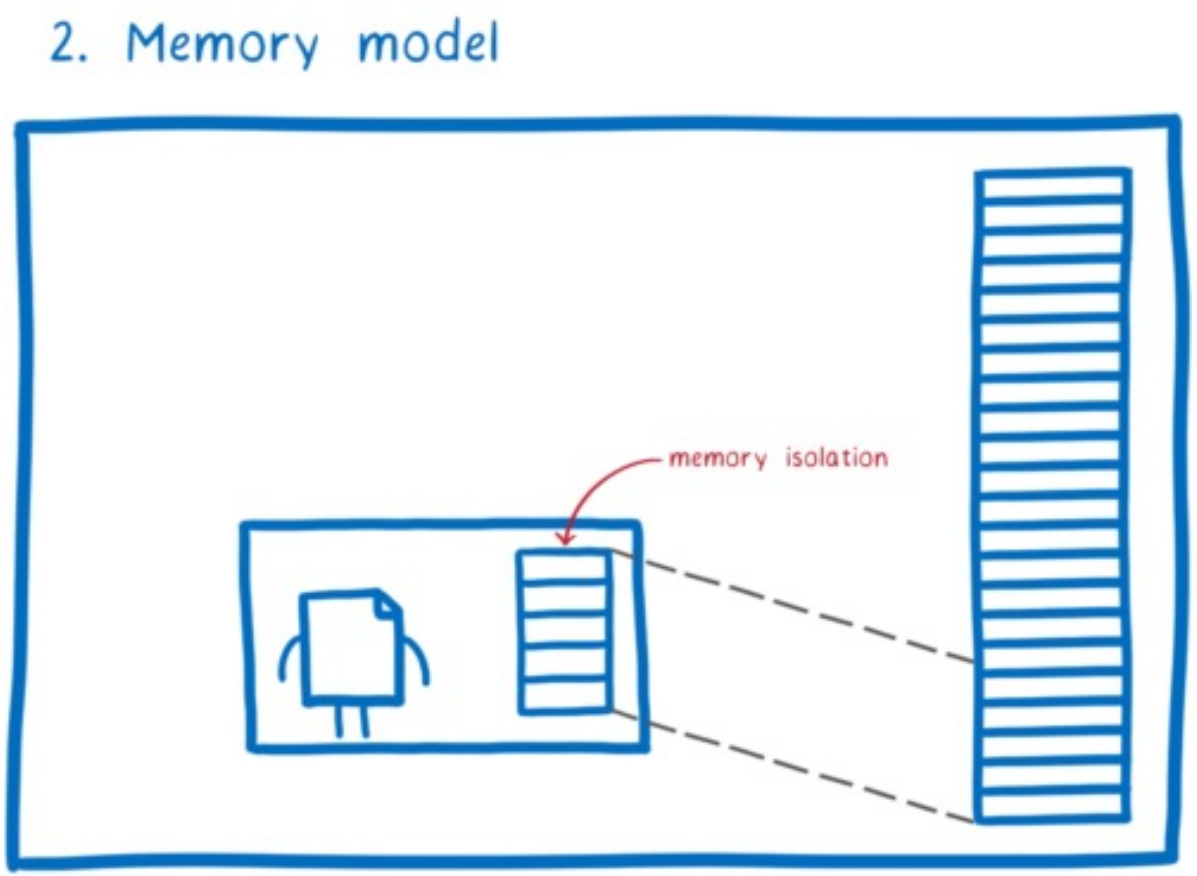
@nielstanis@infosec.exchange

WebAssembly Nano-Process



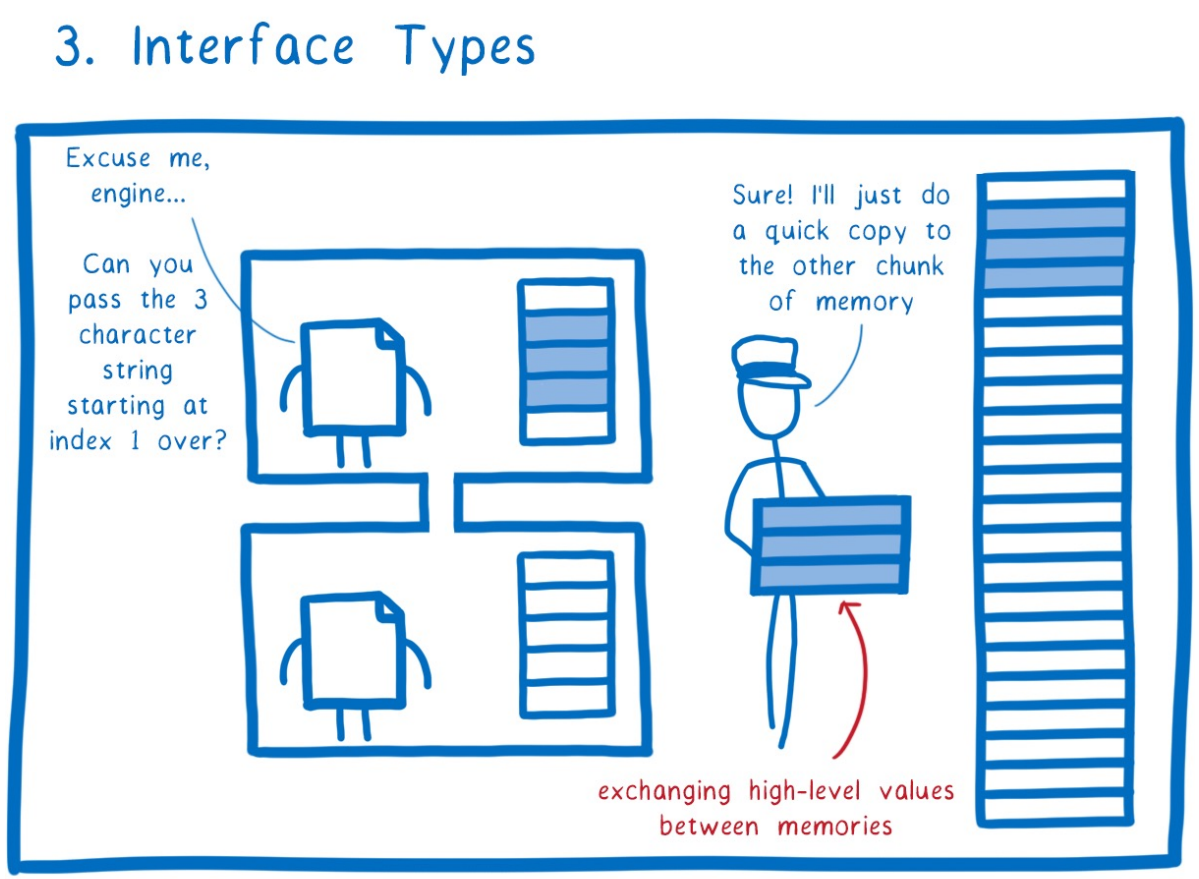
@nielstanis@infosec.exchange

WebAssembly Nano-Process



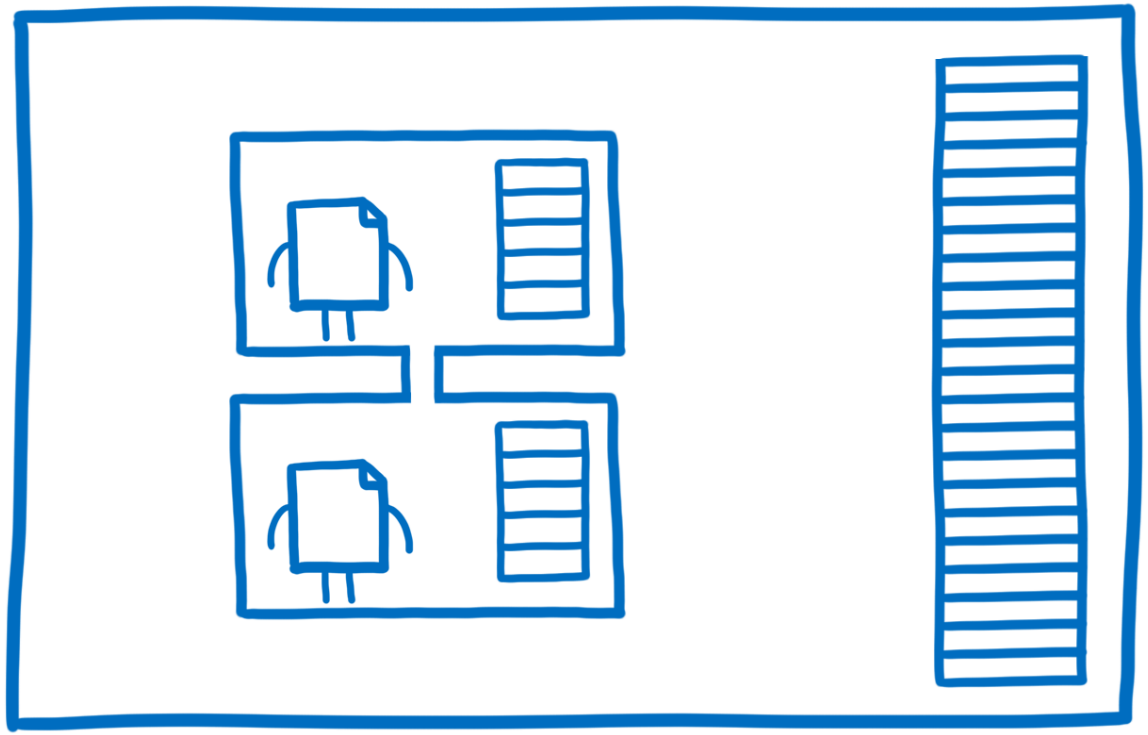
@nielstanis@infosec.exchange

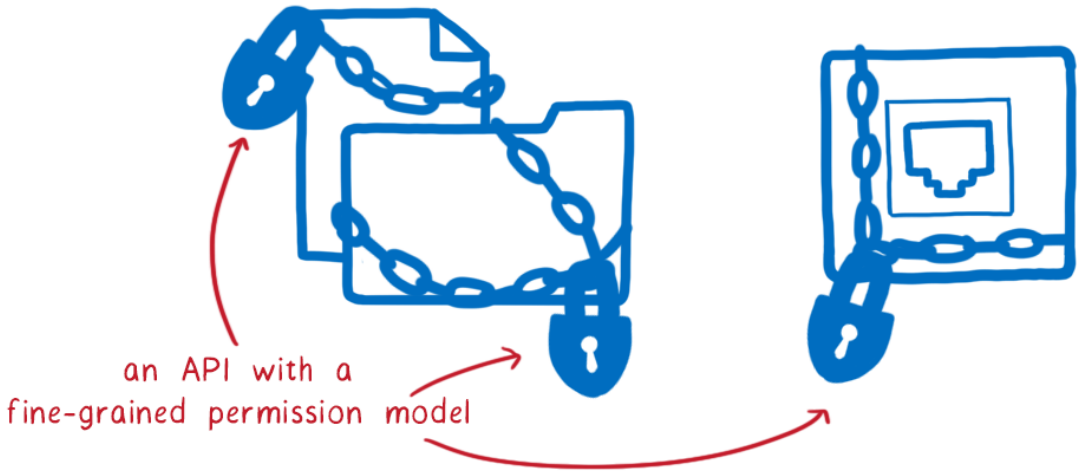
WebAssembly Nano-Process



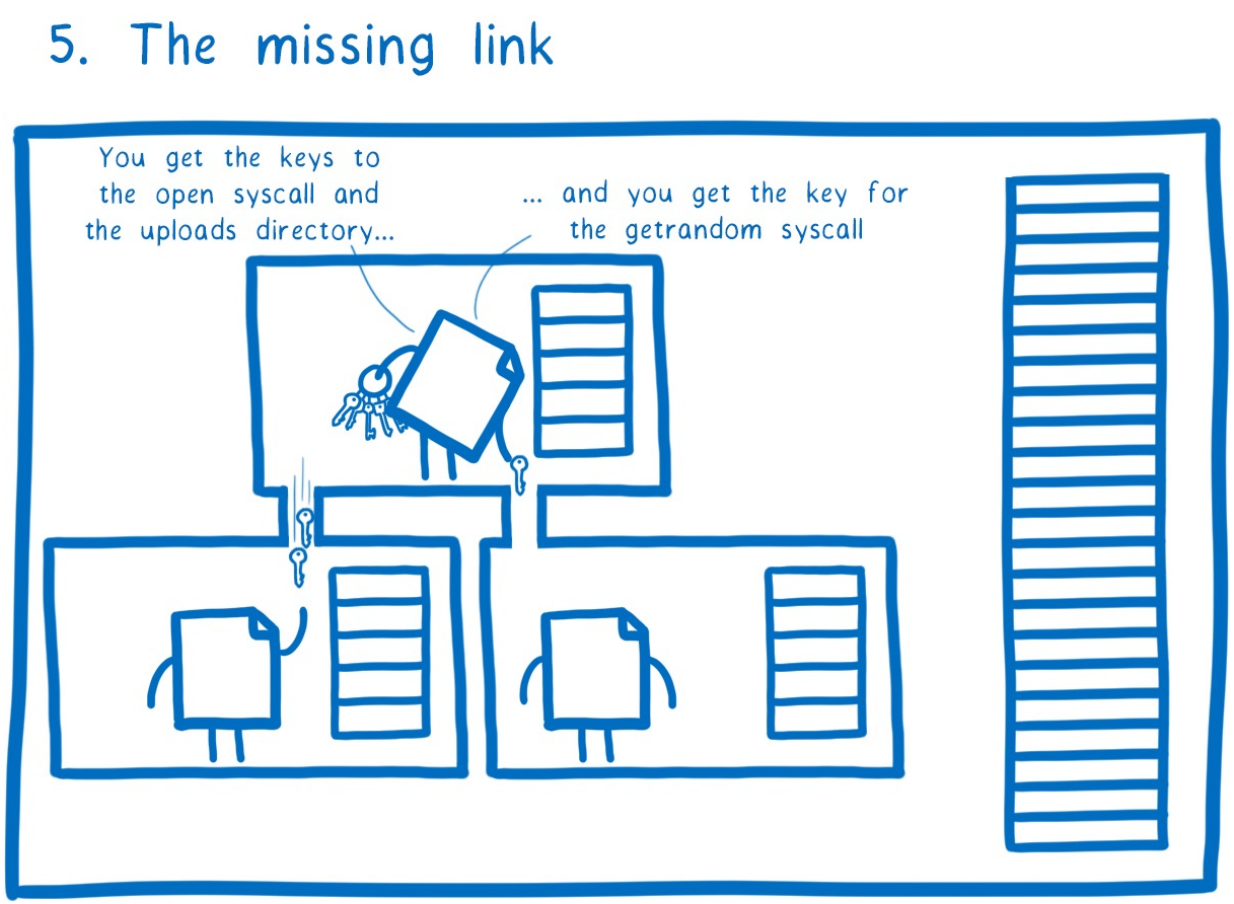
WebAssembly Nano-Process

4. WebAssembly System Interface



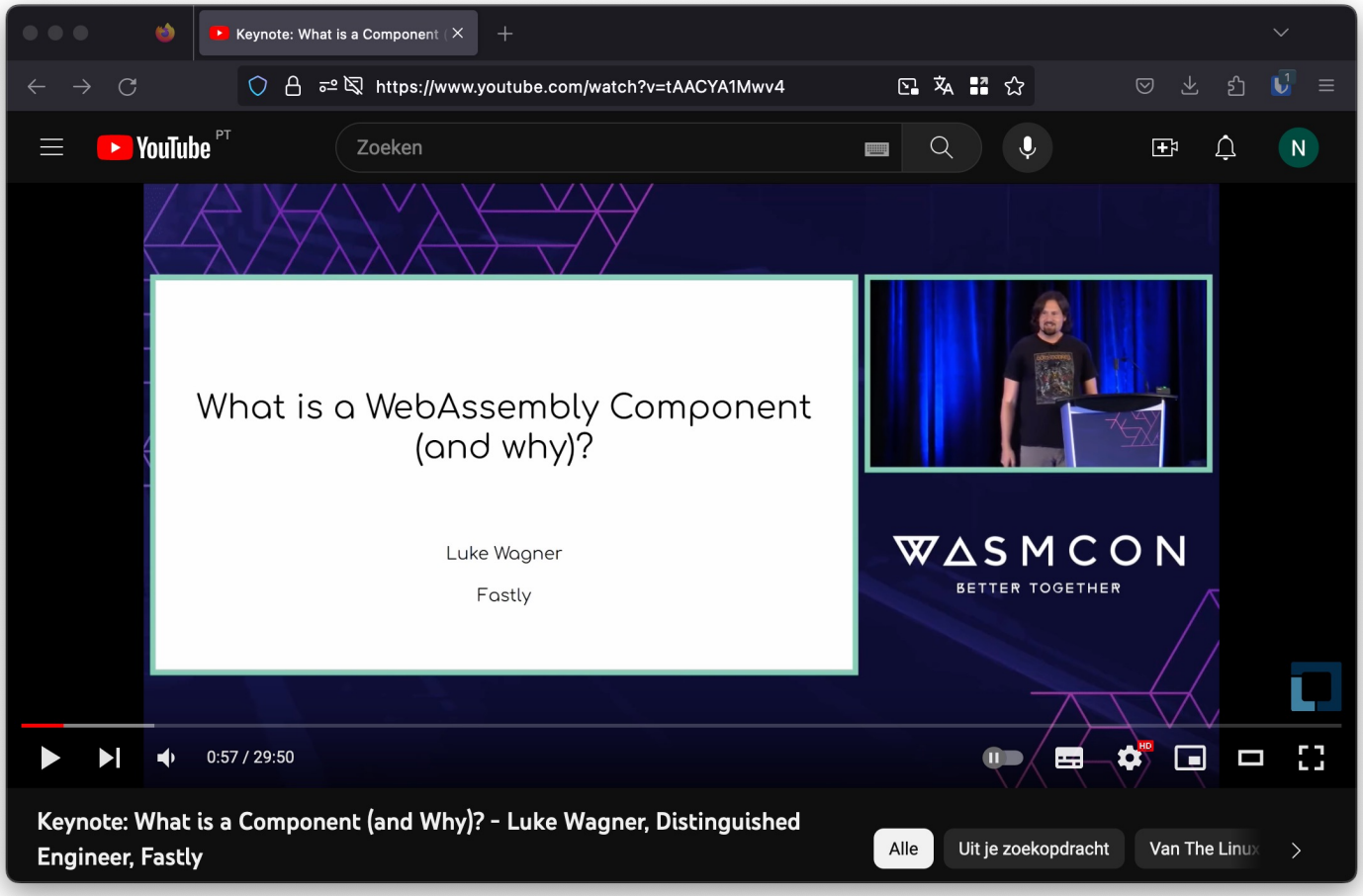


WebAssembly Nano-Process



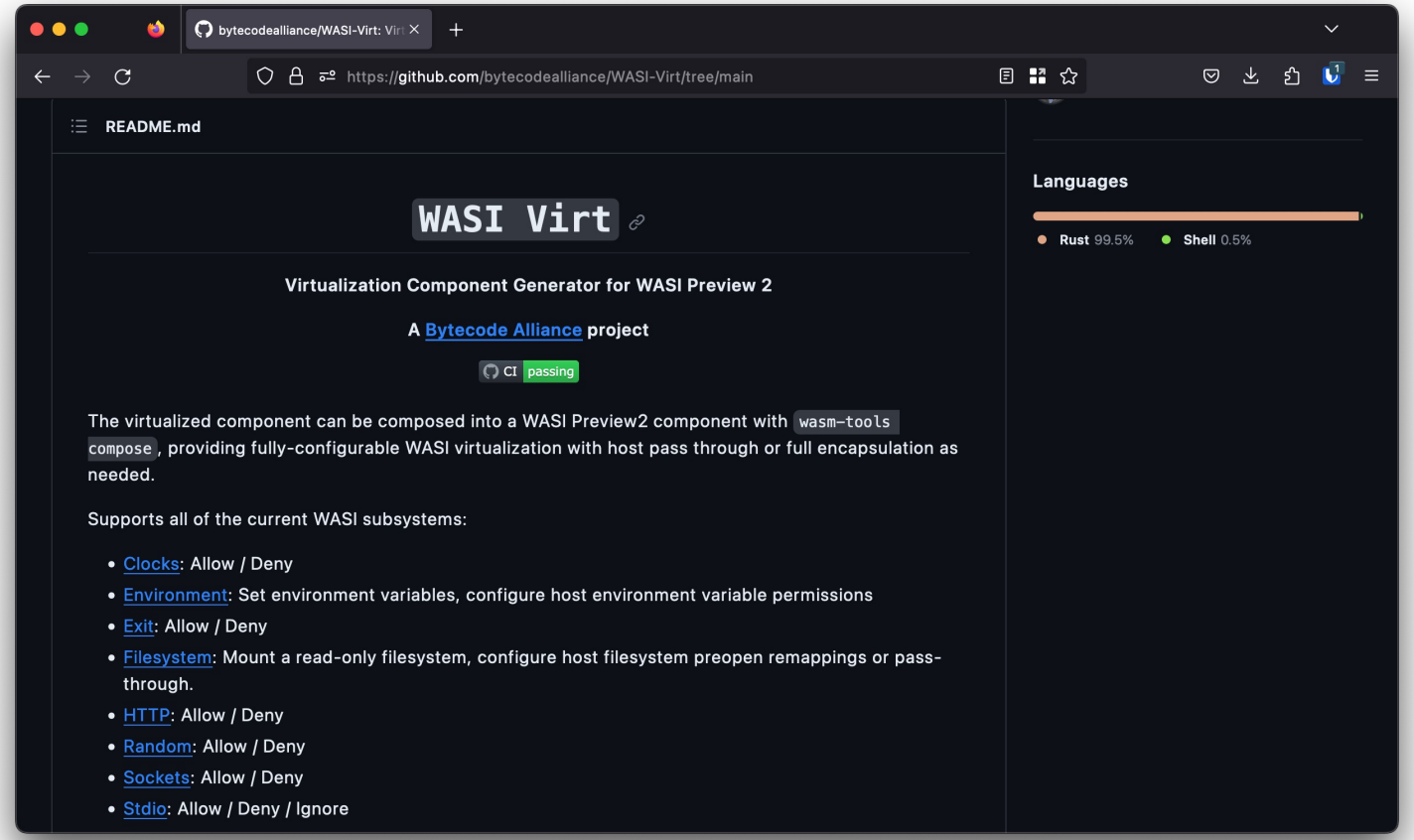
@nielstanis@infosec.exchange

WebAssembly Component Model



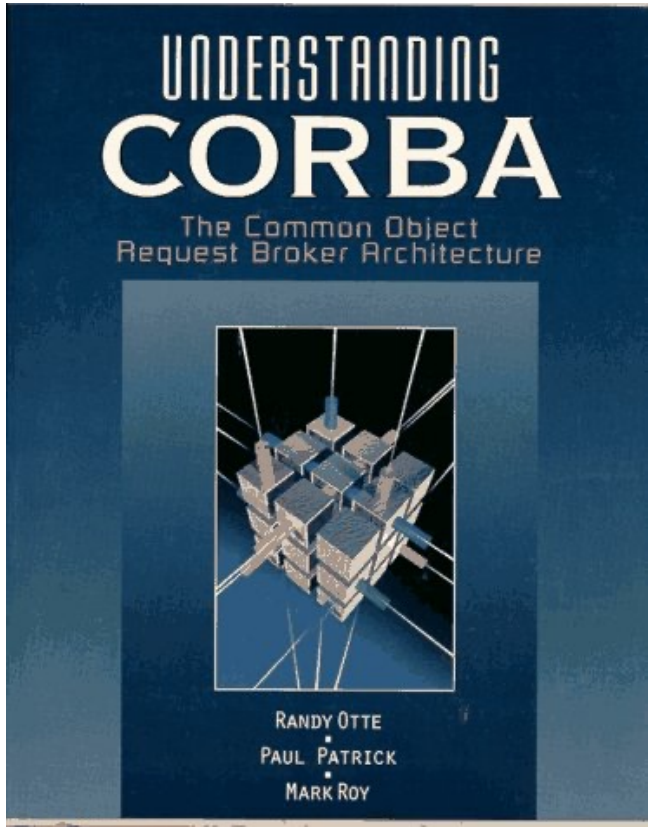
@nielstanis@infosec.exchange

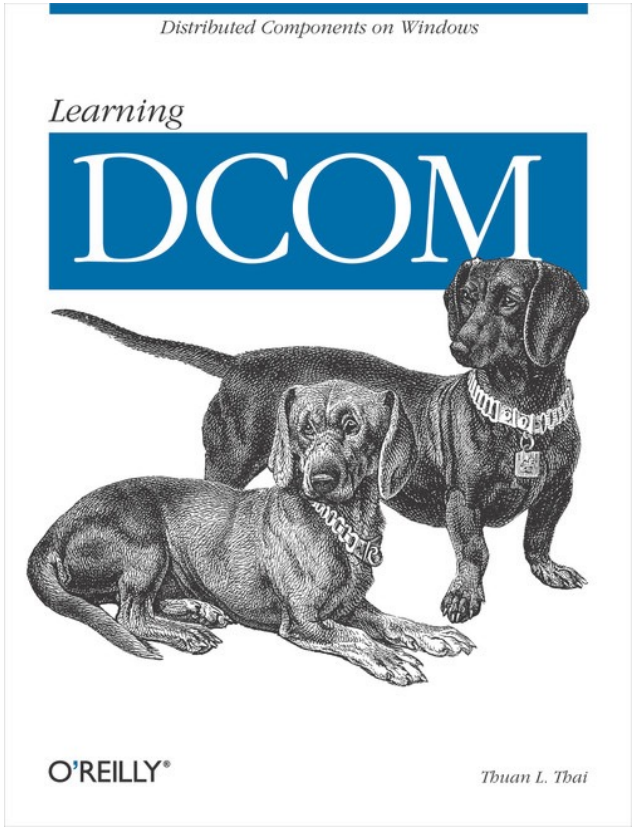
WebAssembly Component Model

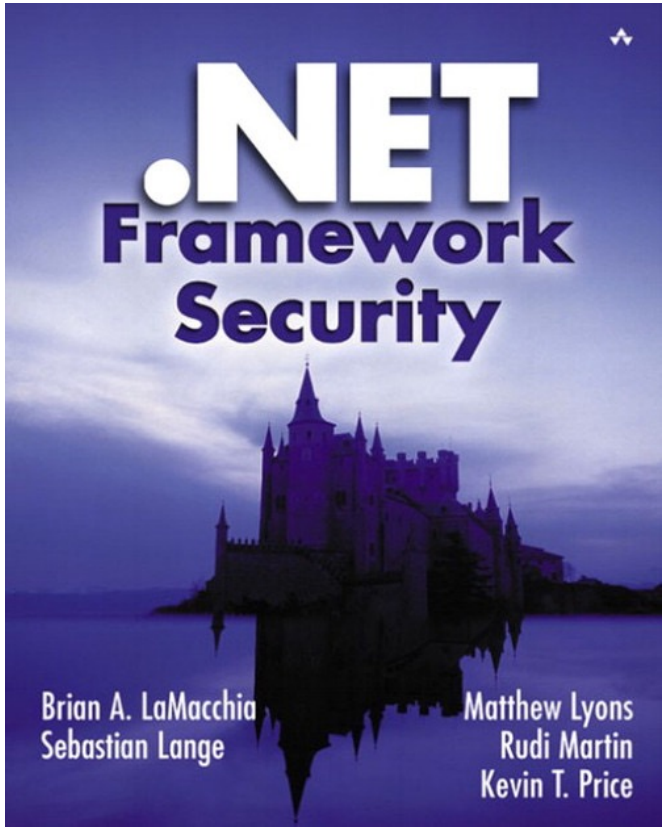


@nielstanis@infosec.exchange

Have we seen this before?







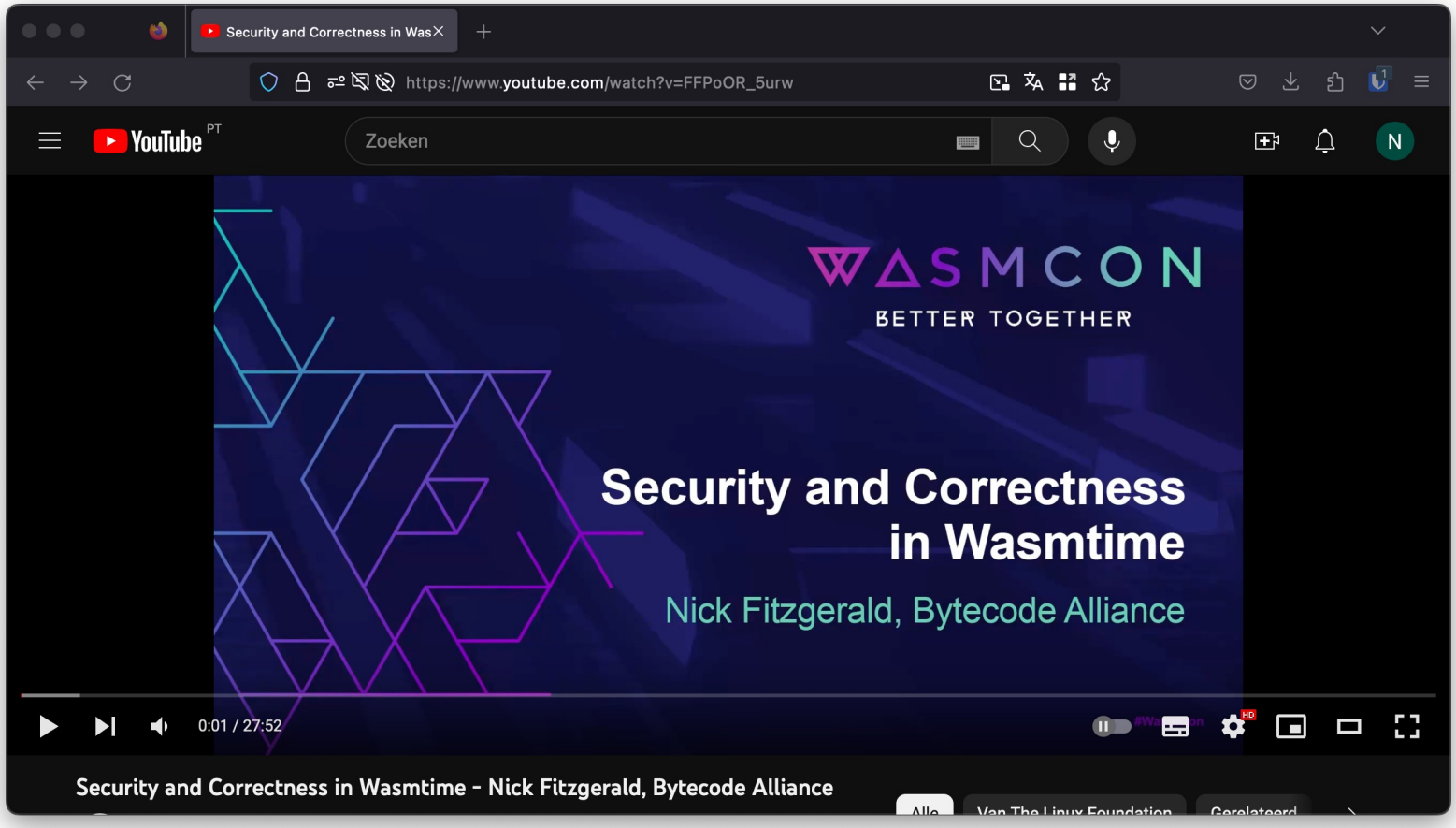
@nielstanis@infosec.exchange

Runtimes and Security

- Most security research published focusses on correctness of WASM runtimes/VM's based on e.g. fuzzing

- Bytecode Alliance Blogpost September 2022 by Nick Fitzgerald:
  - "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



@nielstanis@infosec.exchange

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!
- I like top-down approach Bytecode Alliance is taking in moving forward, feedback will change/influence

Questions?

- https://github.com/nielstanis/owaspbenelux2023
- ntanis at Veracode.com
- @nielstanis@infosec.exchange
- https://blog.fennec.dev
- Merci! Bedankt!