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

Niels Tanis

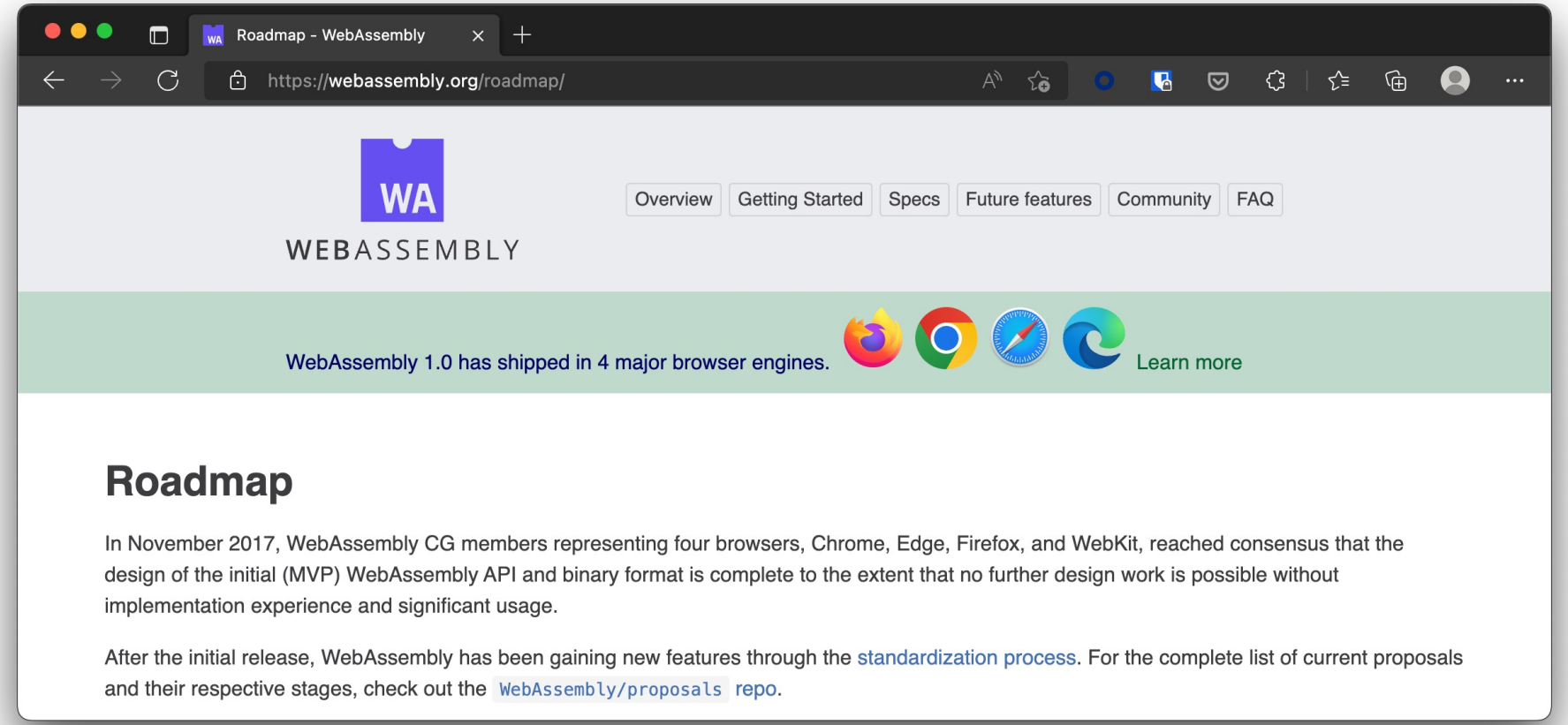
Update Conference
Prague 2022

Who am I?

- Niels Tanis
- Sr. Principal Security Researcher @ Veracode
  - Background .NET Development, Pentesting/ethical hacking, and software security consultancy
  - Research on static analysis for .NET apps

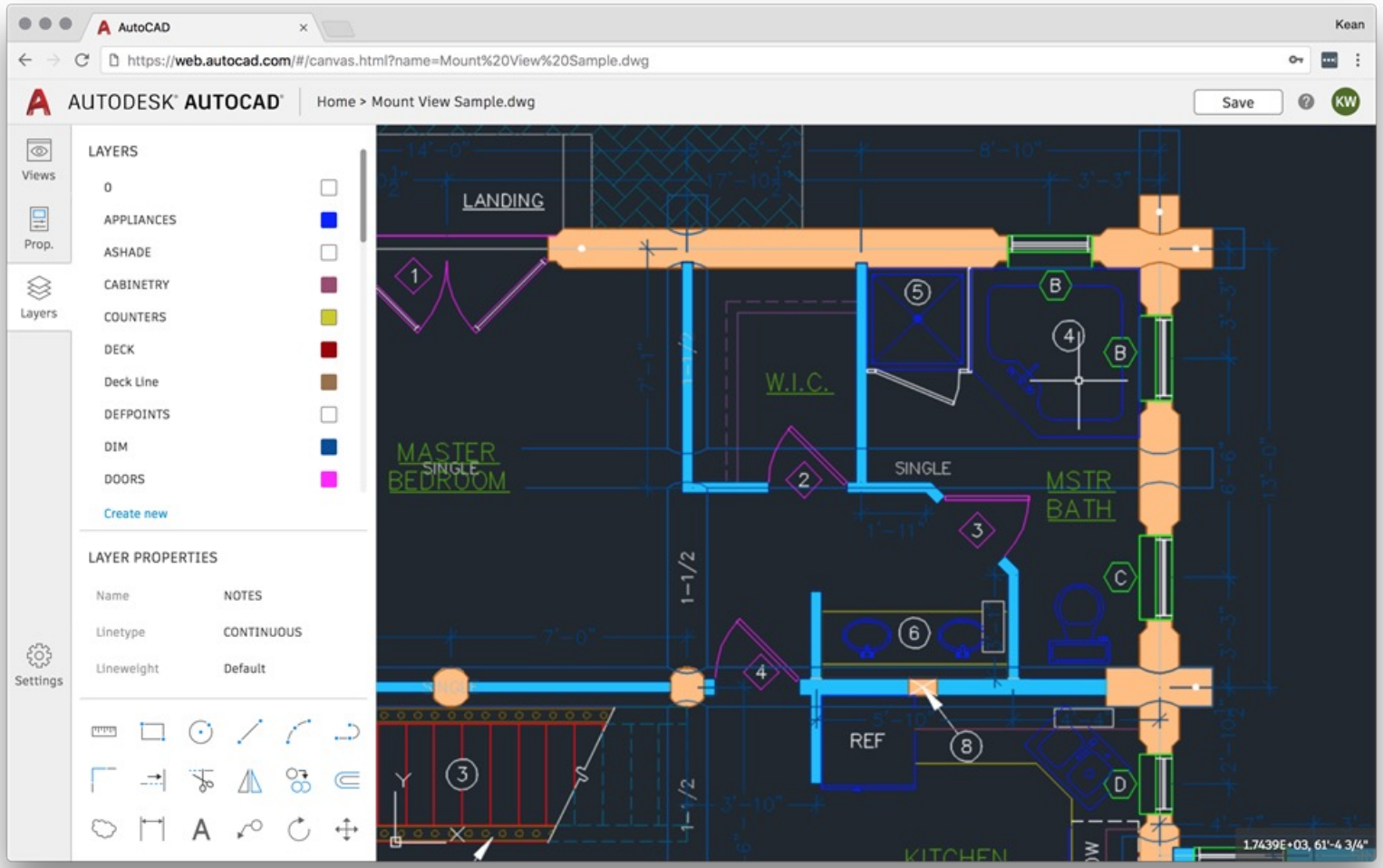
update

WebAssembly



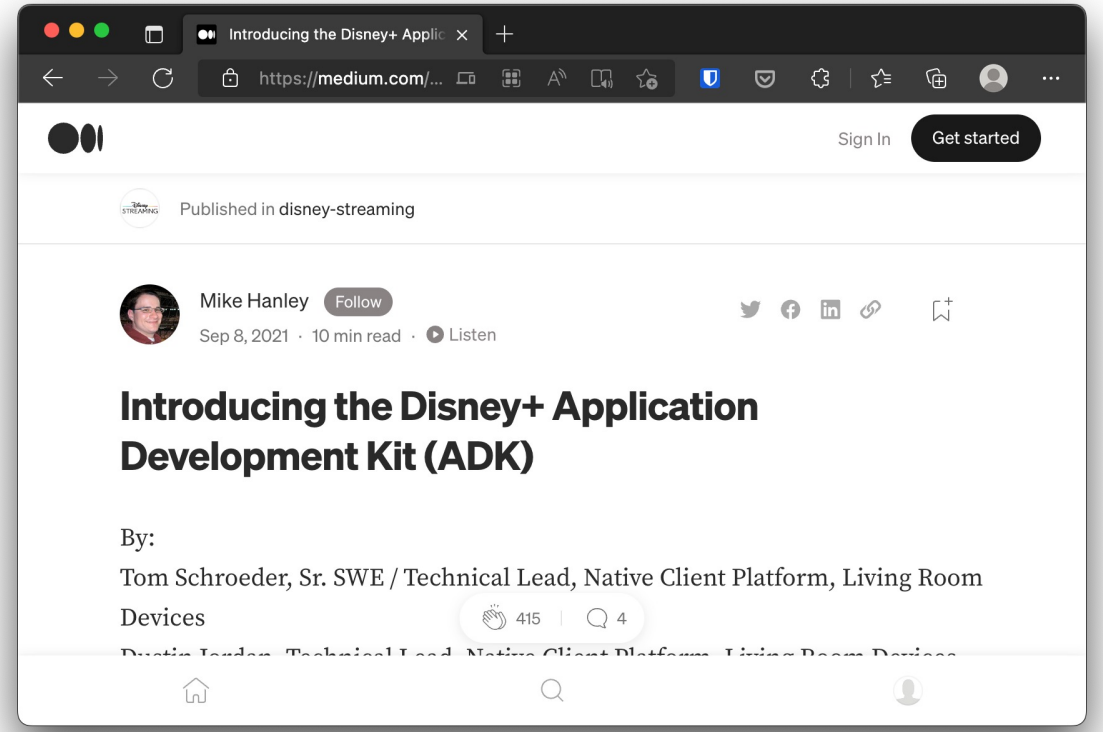
update

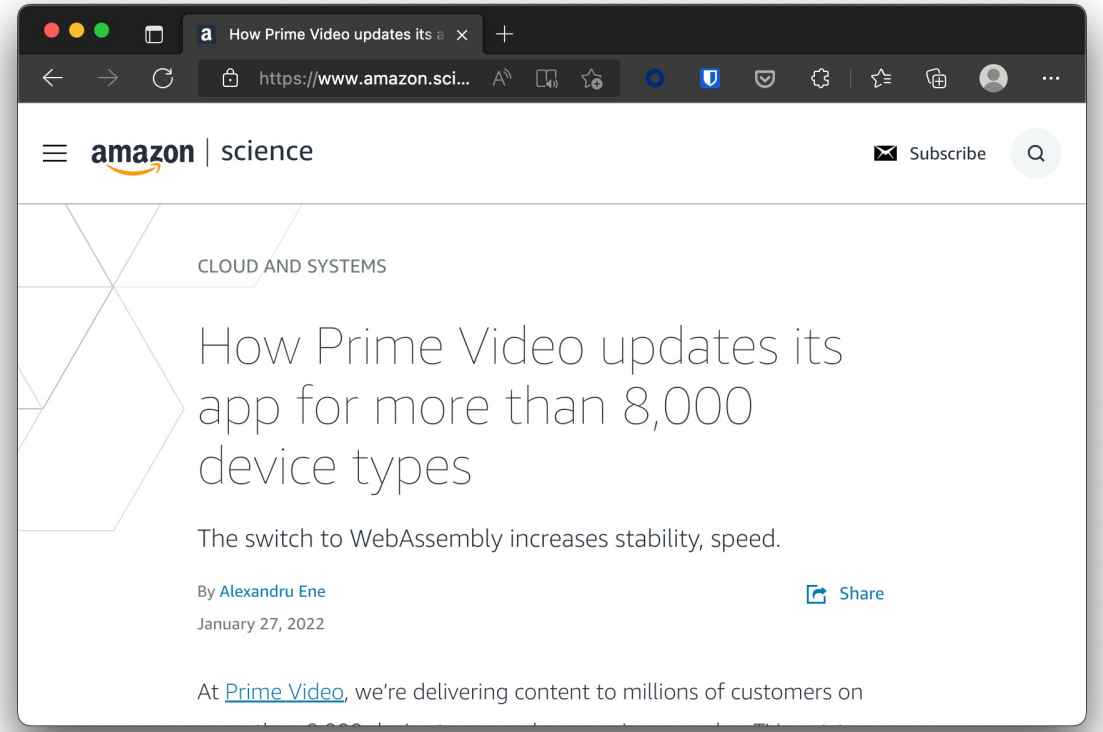
WebAssembly - AutoCAD



update

WebAssembly - SDK's





update

Agenda

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

update

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

update

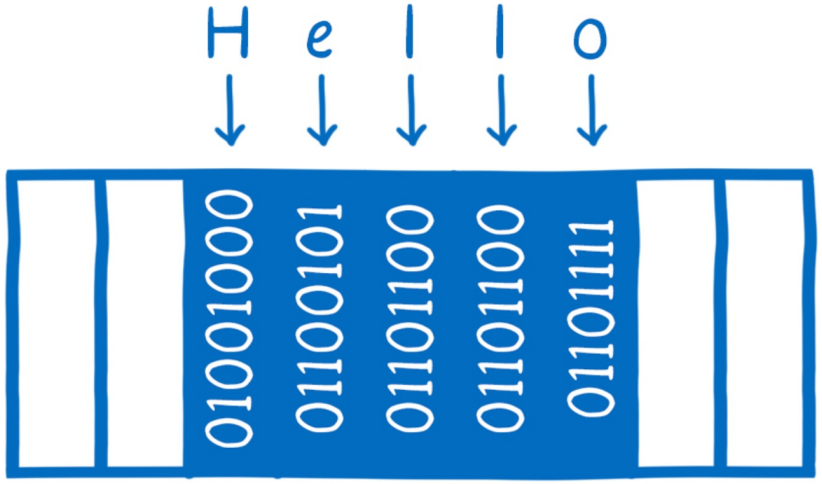
WebAssembly

- 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

update

WebAssembly Memory

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



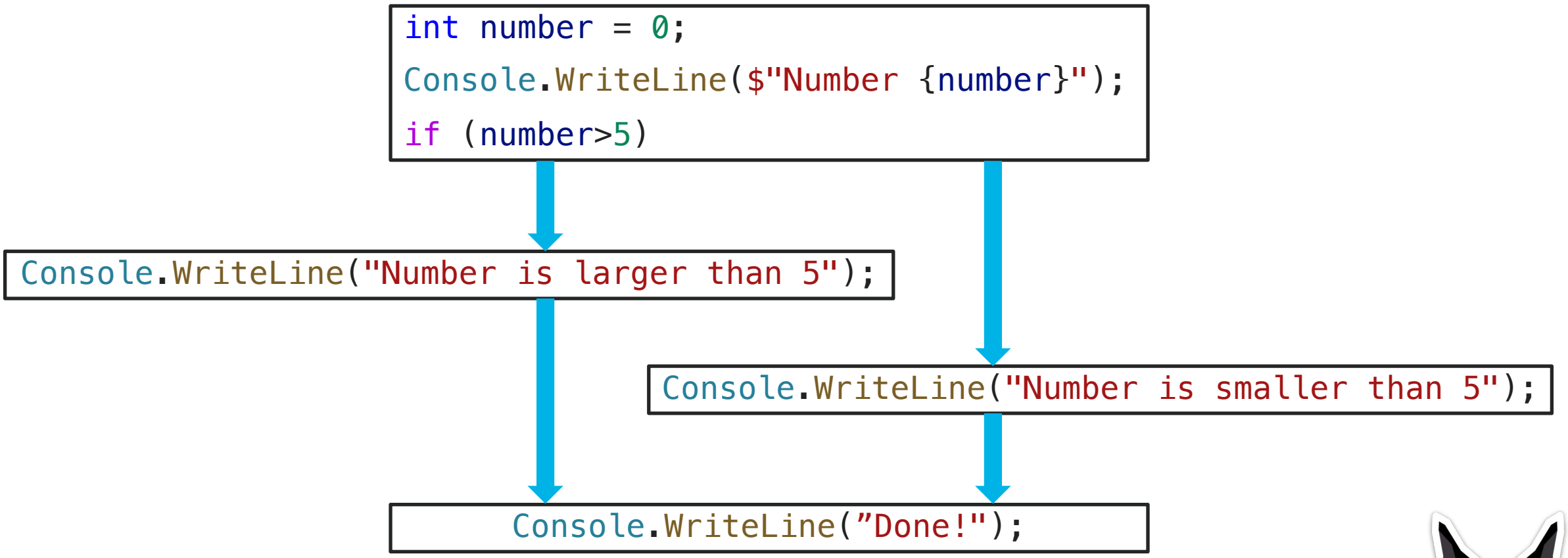
update

WebAssembly Control-Flow Integrity

```
int number = 0;
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!");
```

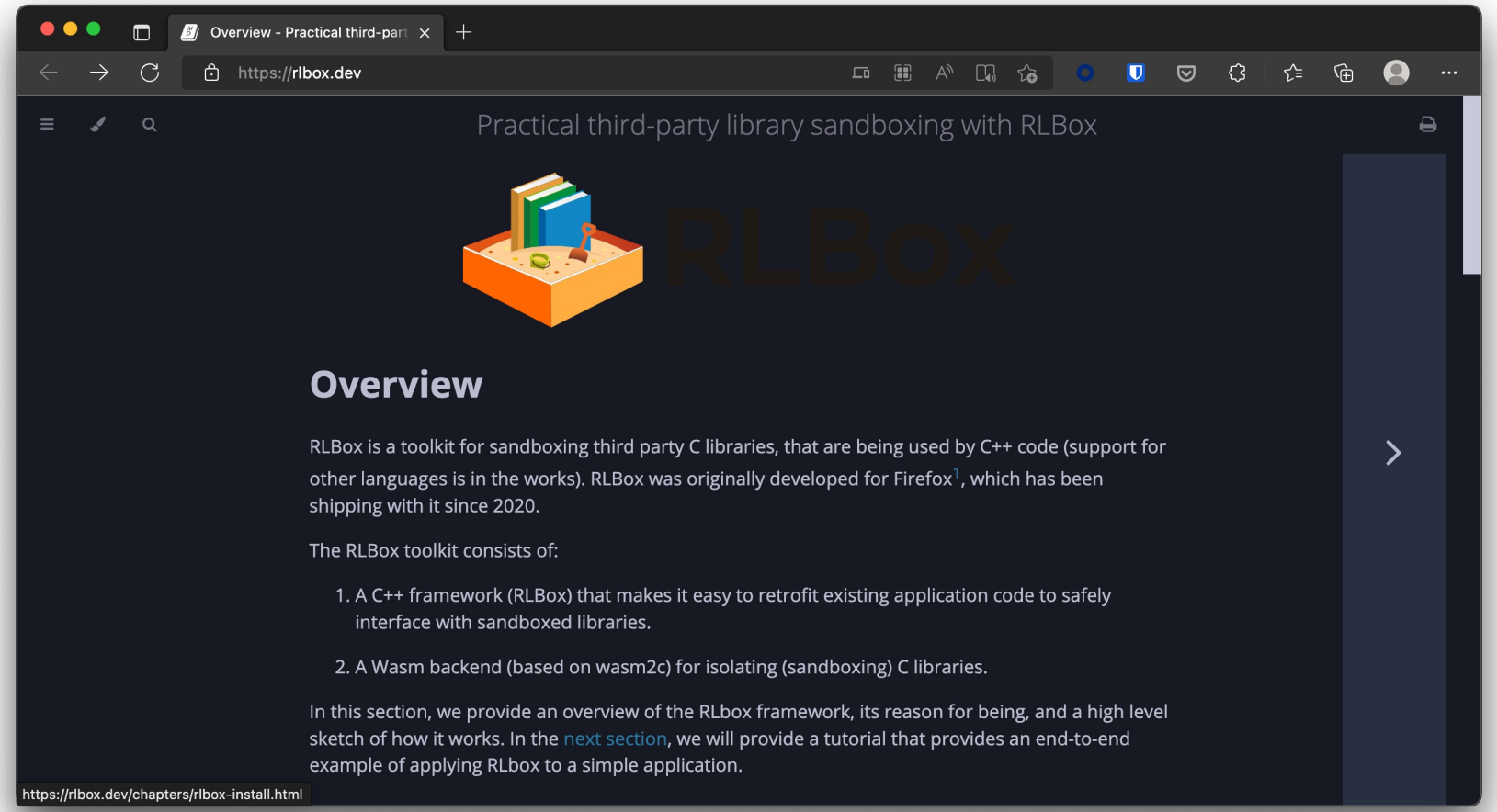
update

WebAssembly Control-Flow Integrity



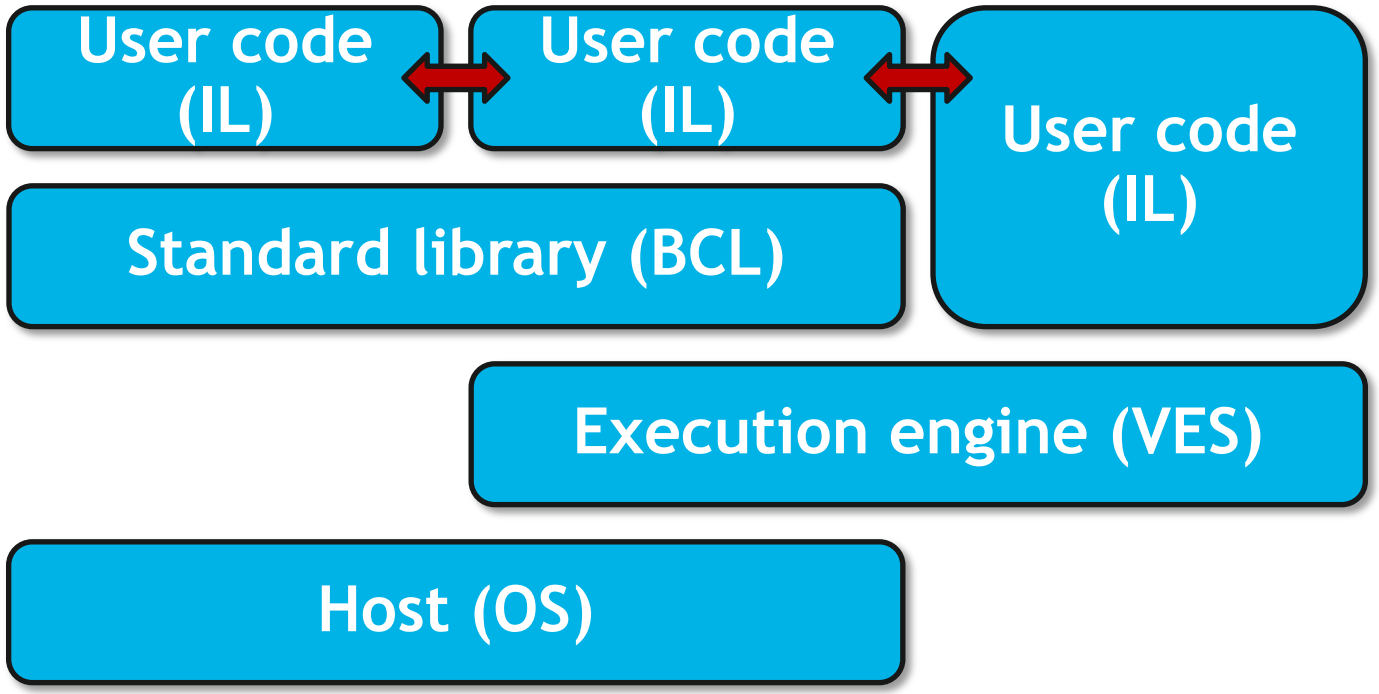
update

FireFox RLBox

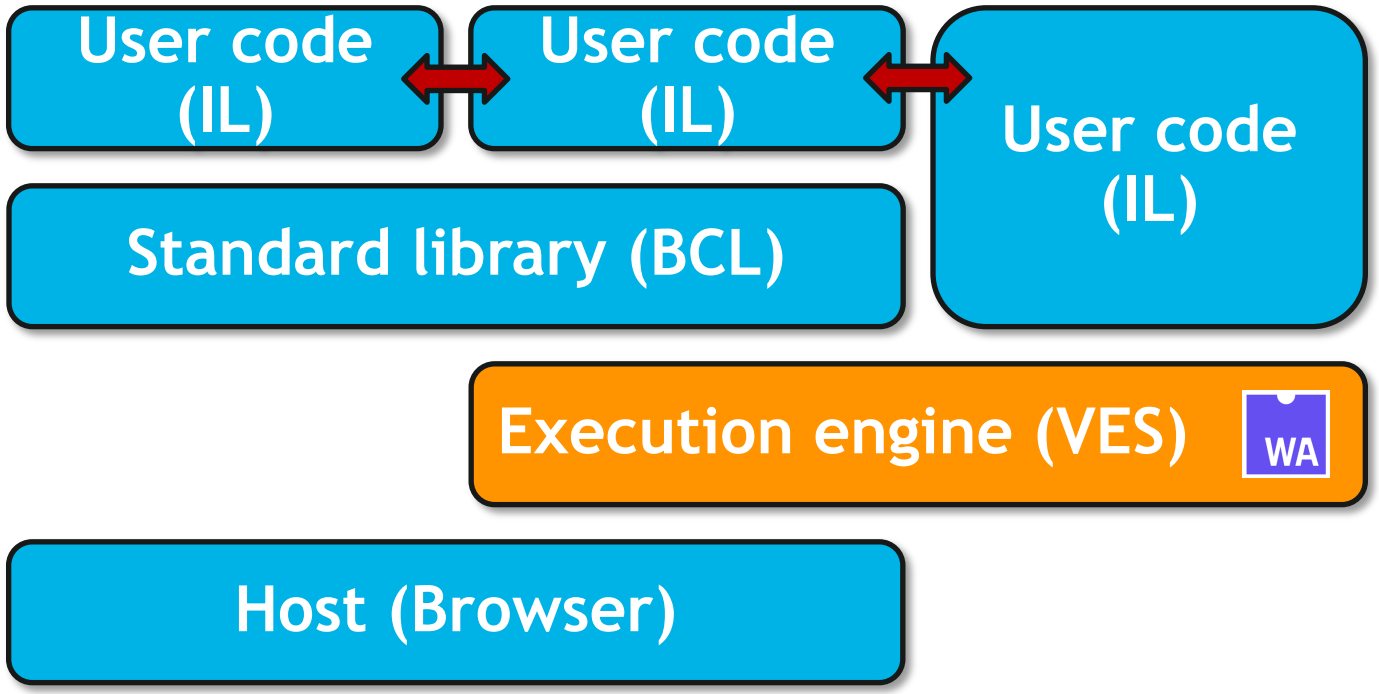


update

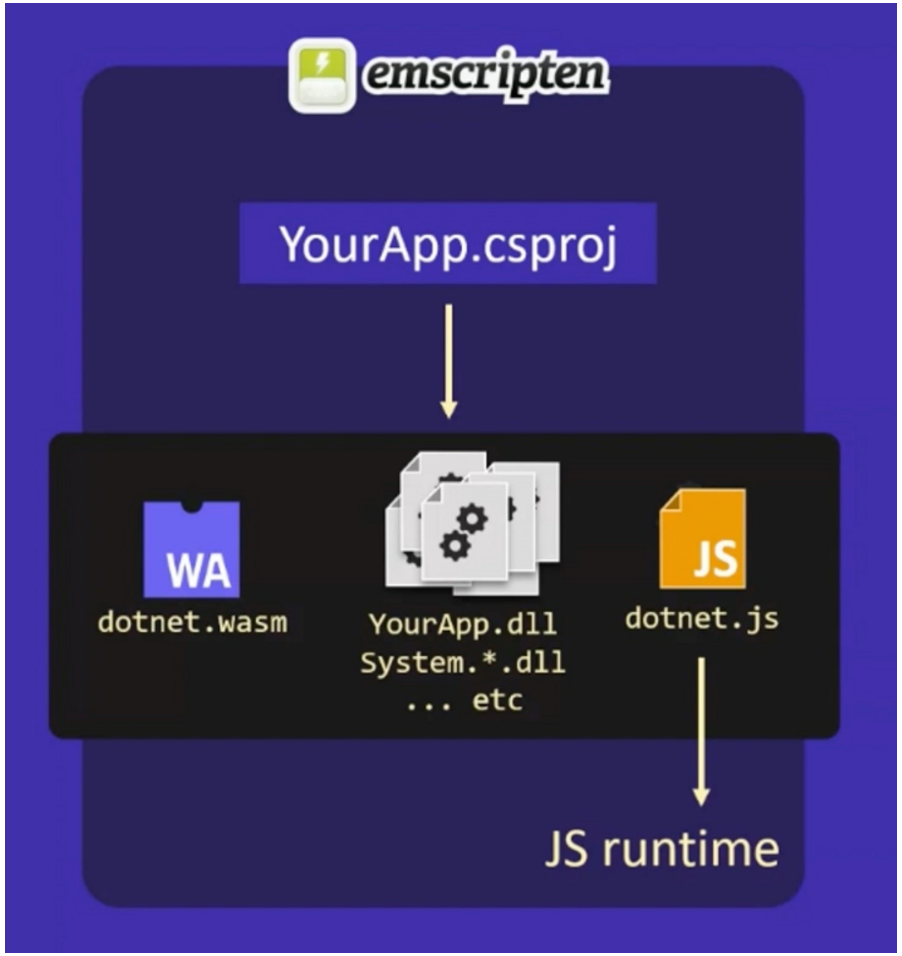
Running .NET on WebAssembly



Running .NET on WebAssembly

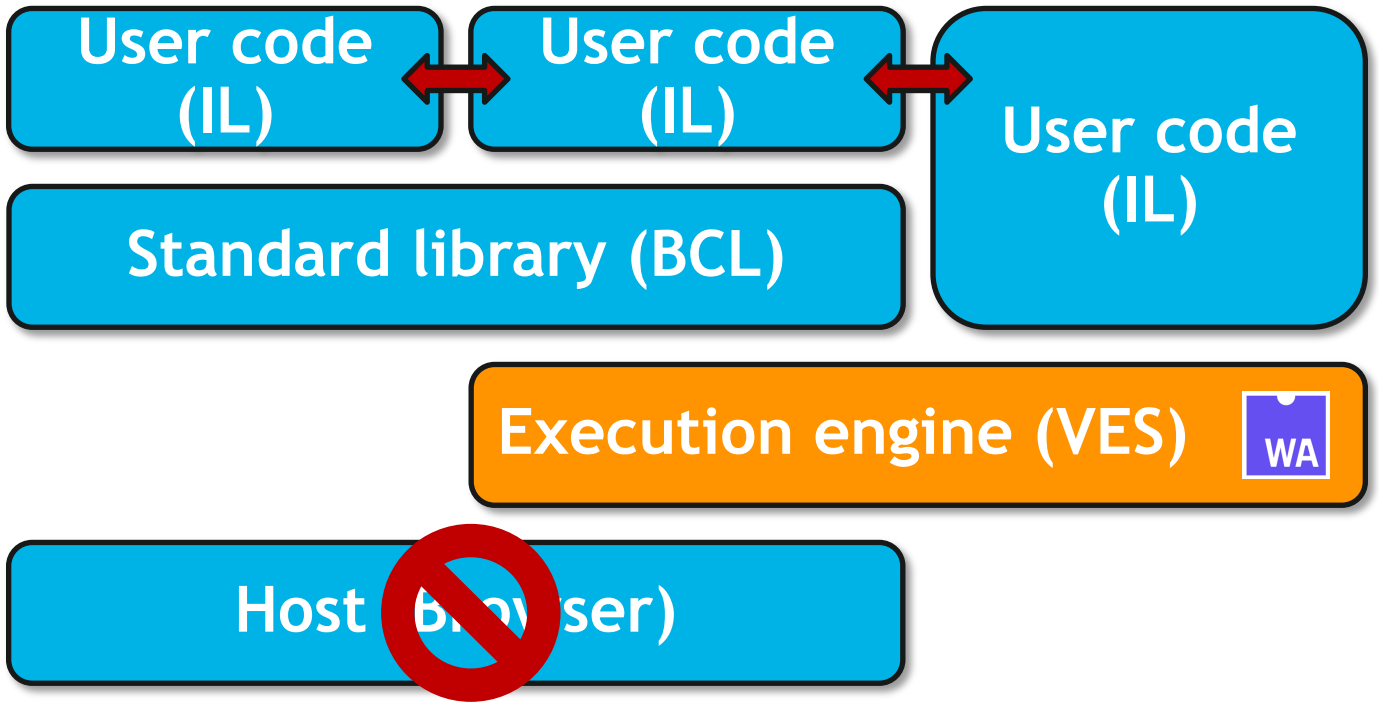


Blazor WebAssembly



update

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

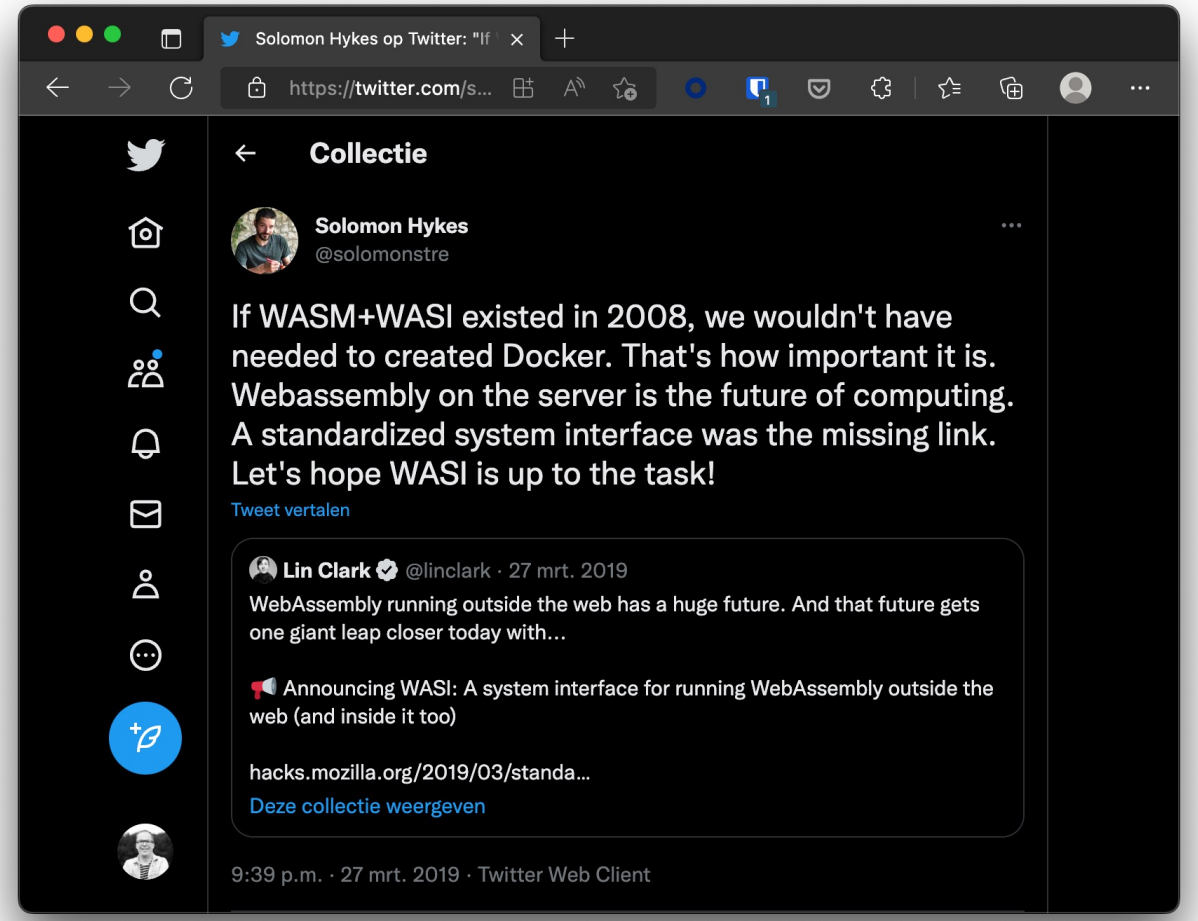
update

WebAssembly System Interface WASI

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

update

Docker vs WASM & WASI

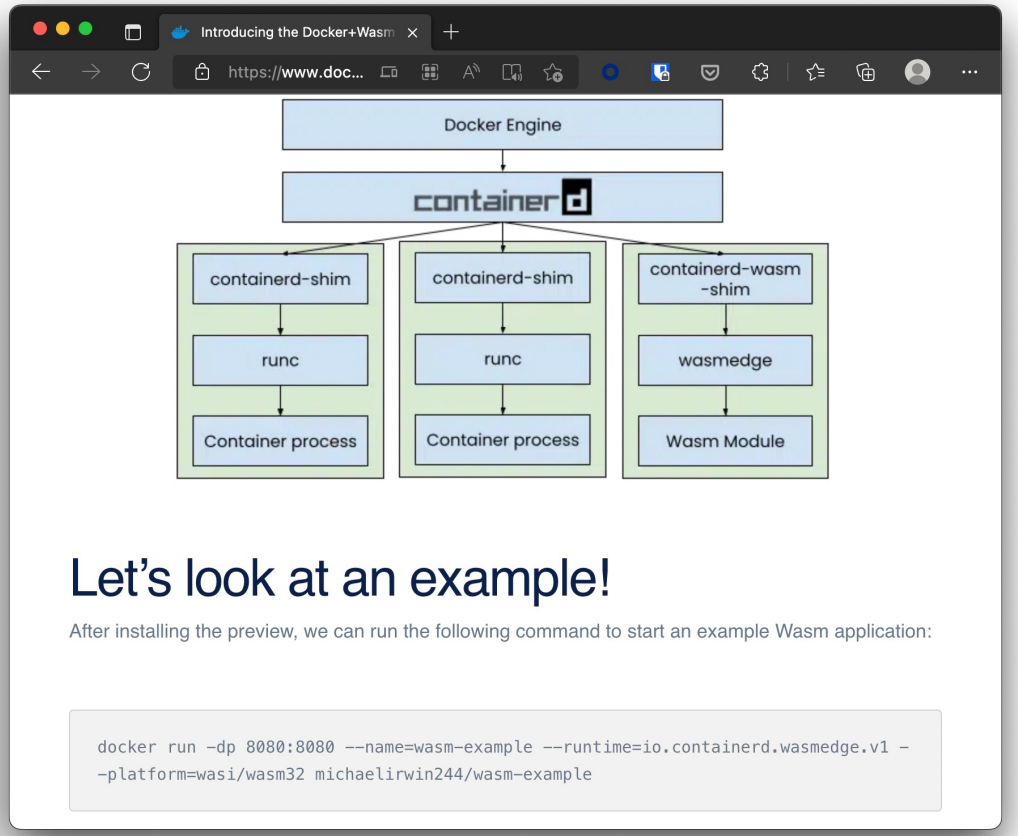


update

Docker vs WASM & WASI

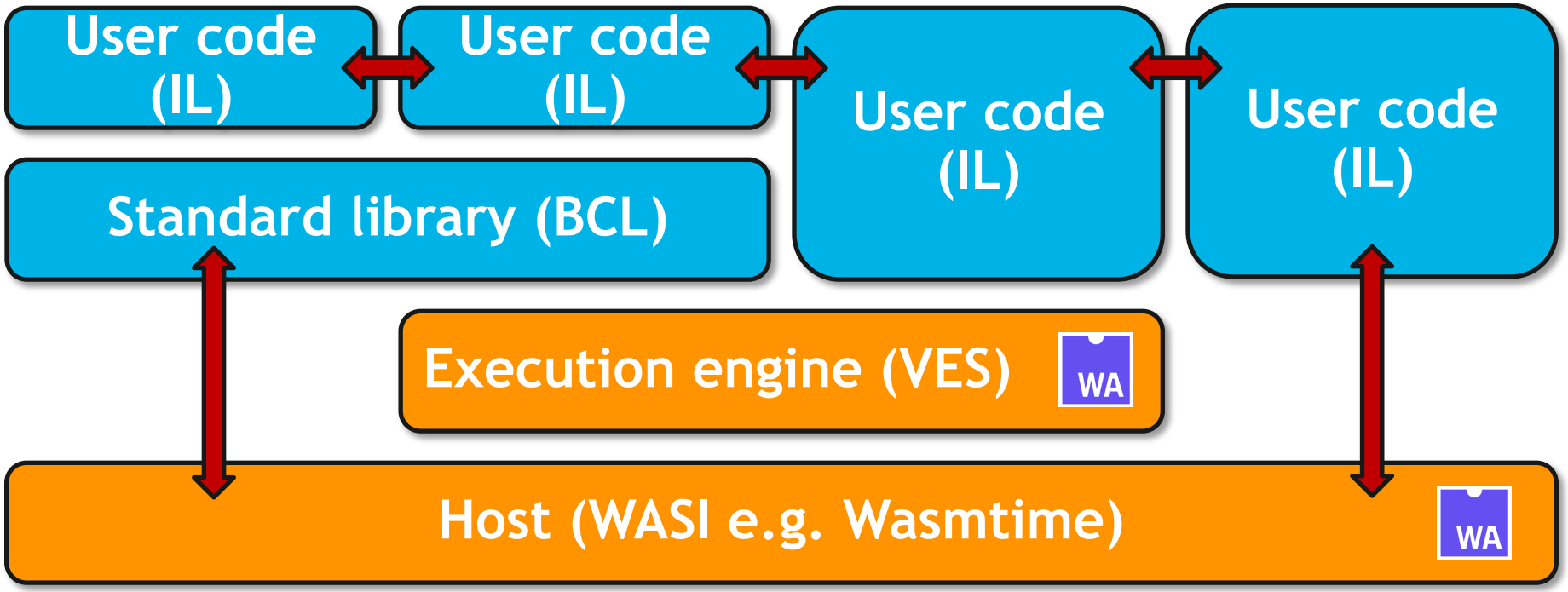
update

Docker & WASM



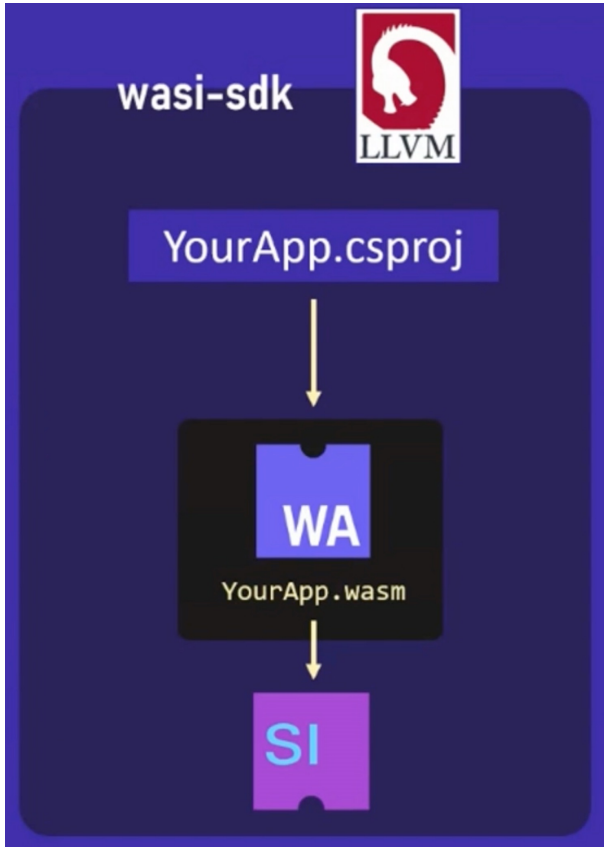
update

WebAssembly System Interface WASI



update

Experimental WASI SDK for .NET



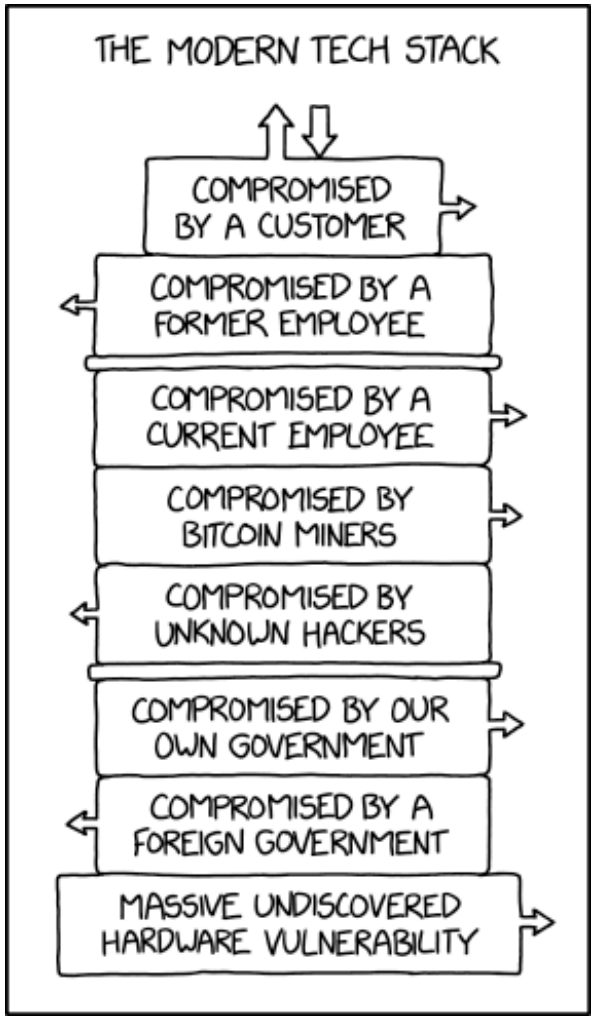
update

Extending .NET with WASM

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

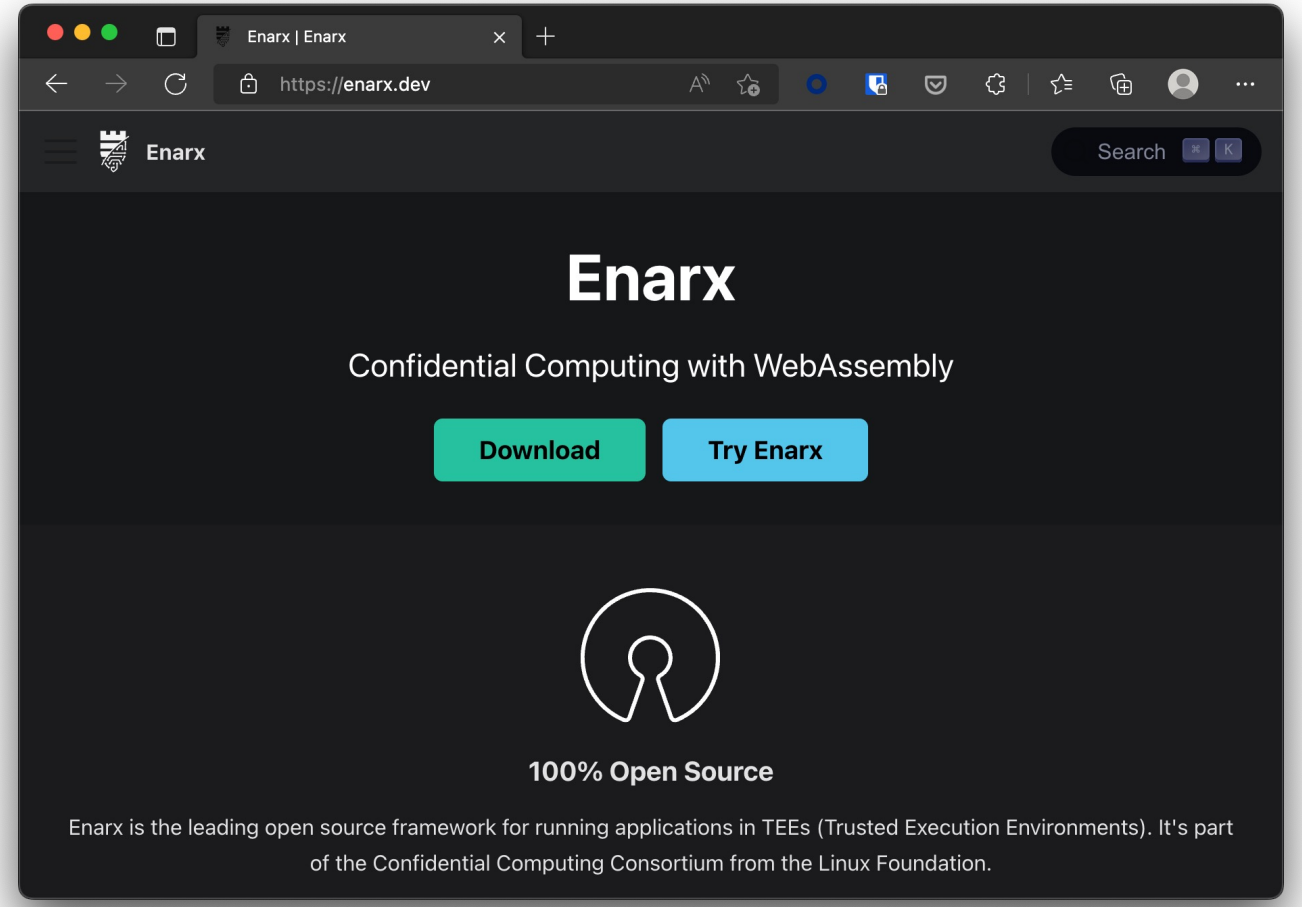
update

Trusted Computing - XKCD 2166



update

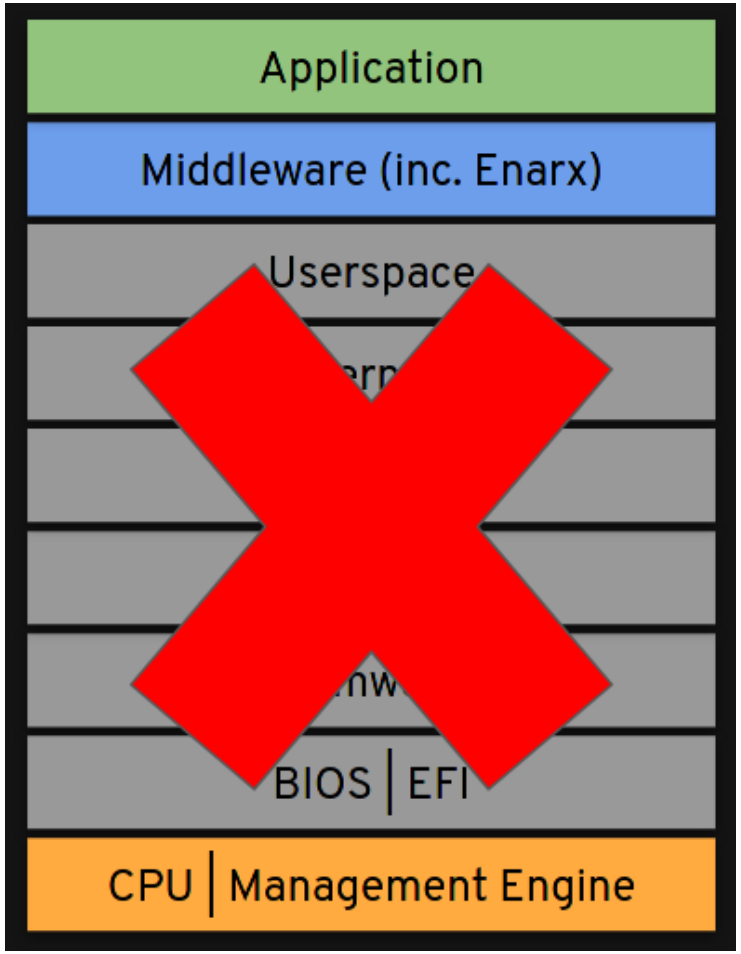
Enarx



update

Enarx Threat Model

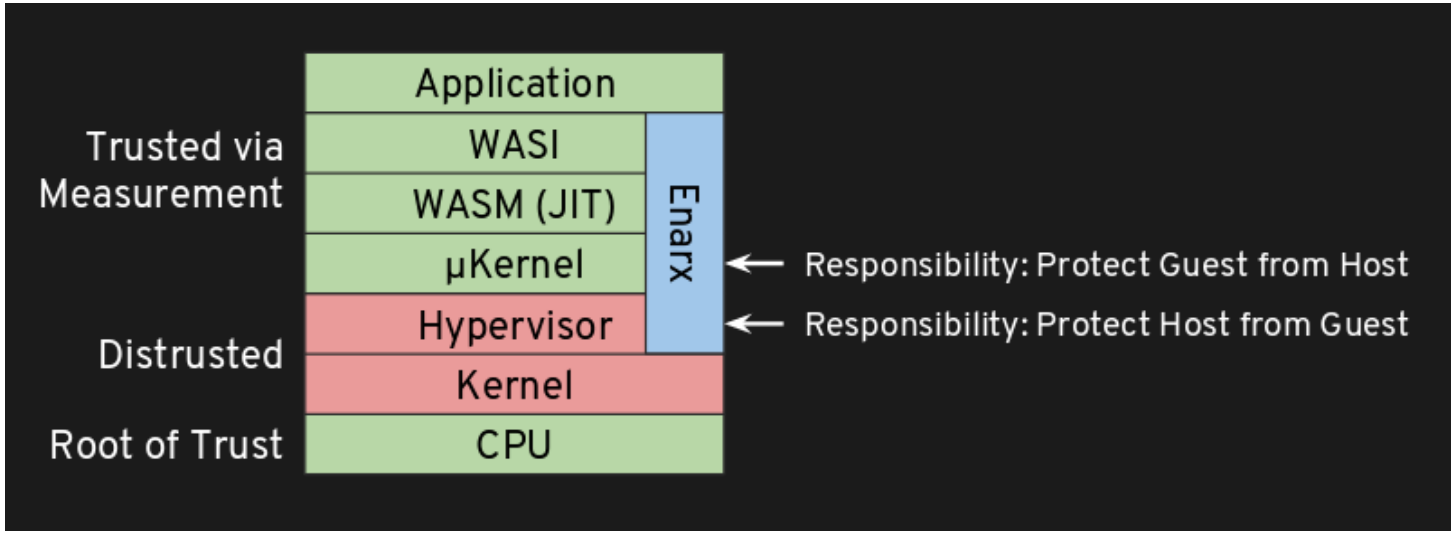
- Don’t trust the host
- Don’t trust the host owner
- Don’t trust the host operator
- Hardware cryptographically verified
- Software audited and cryptographically verified



update

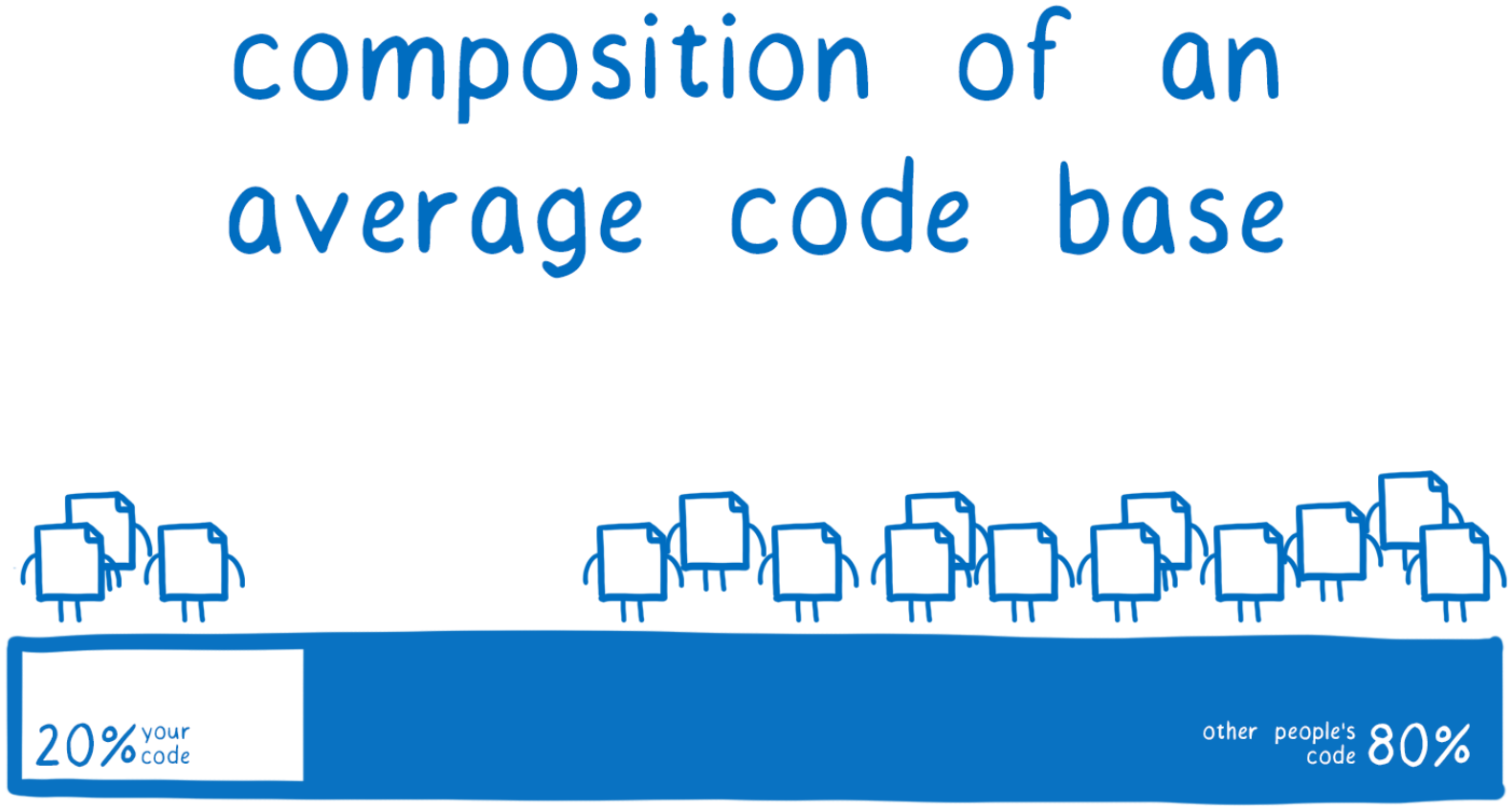
Enarx

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



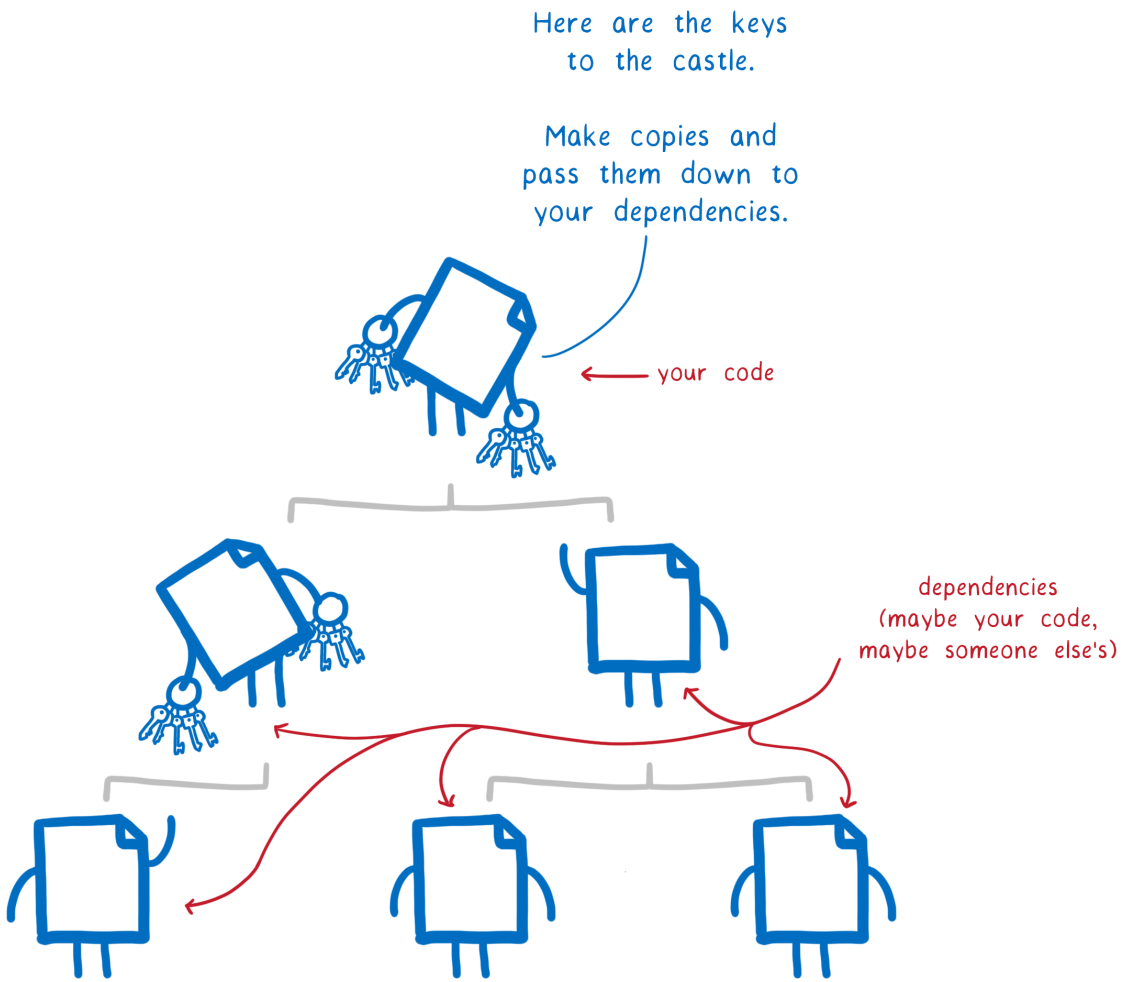
update

WASM – What's next?



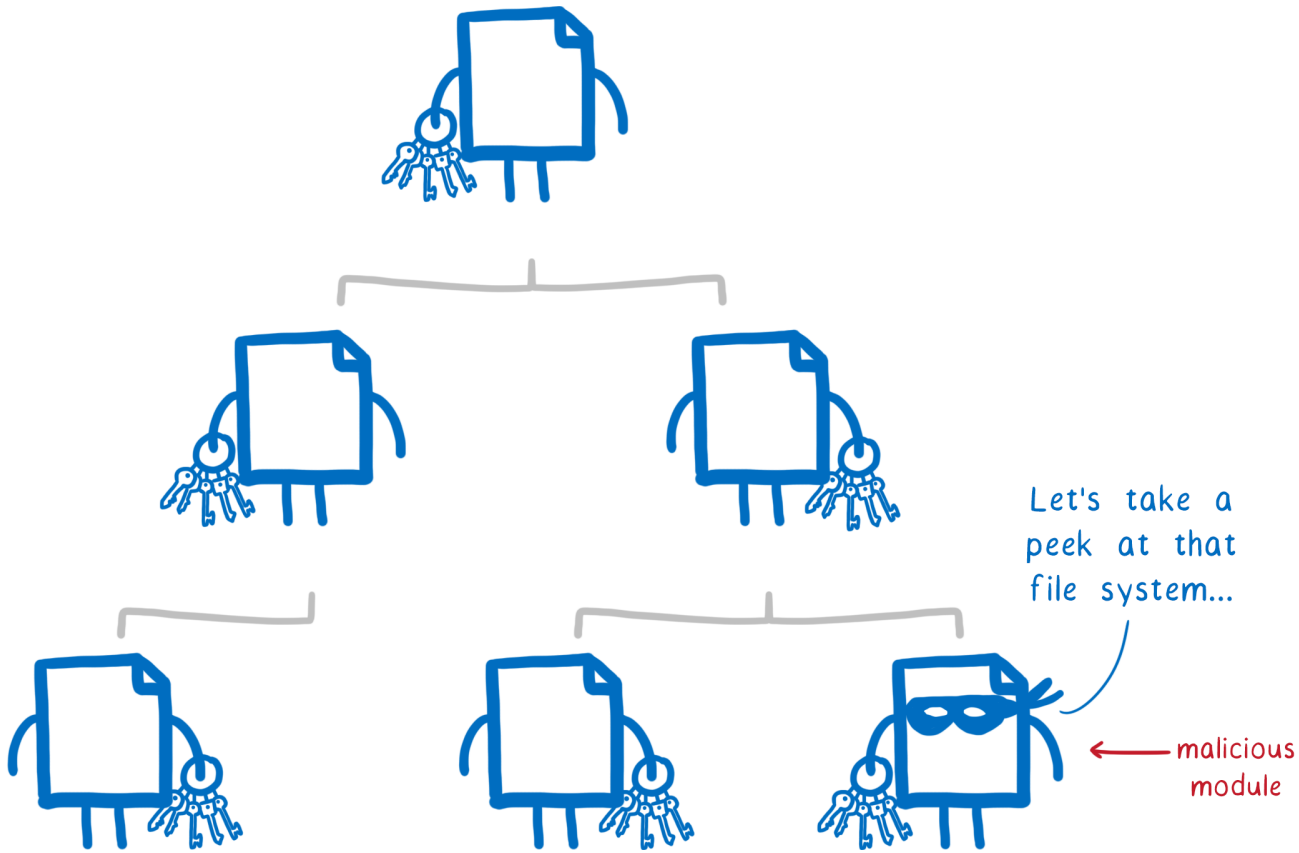
update

Dependencies



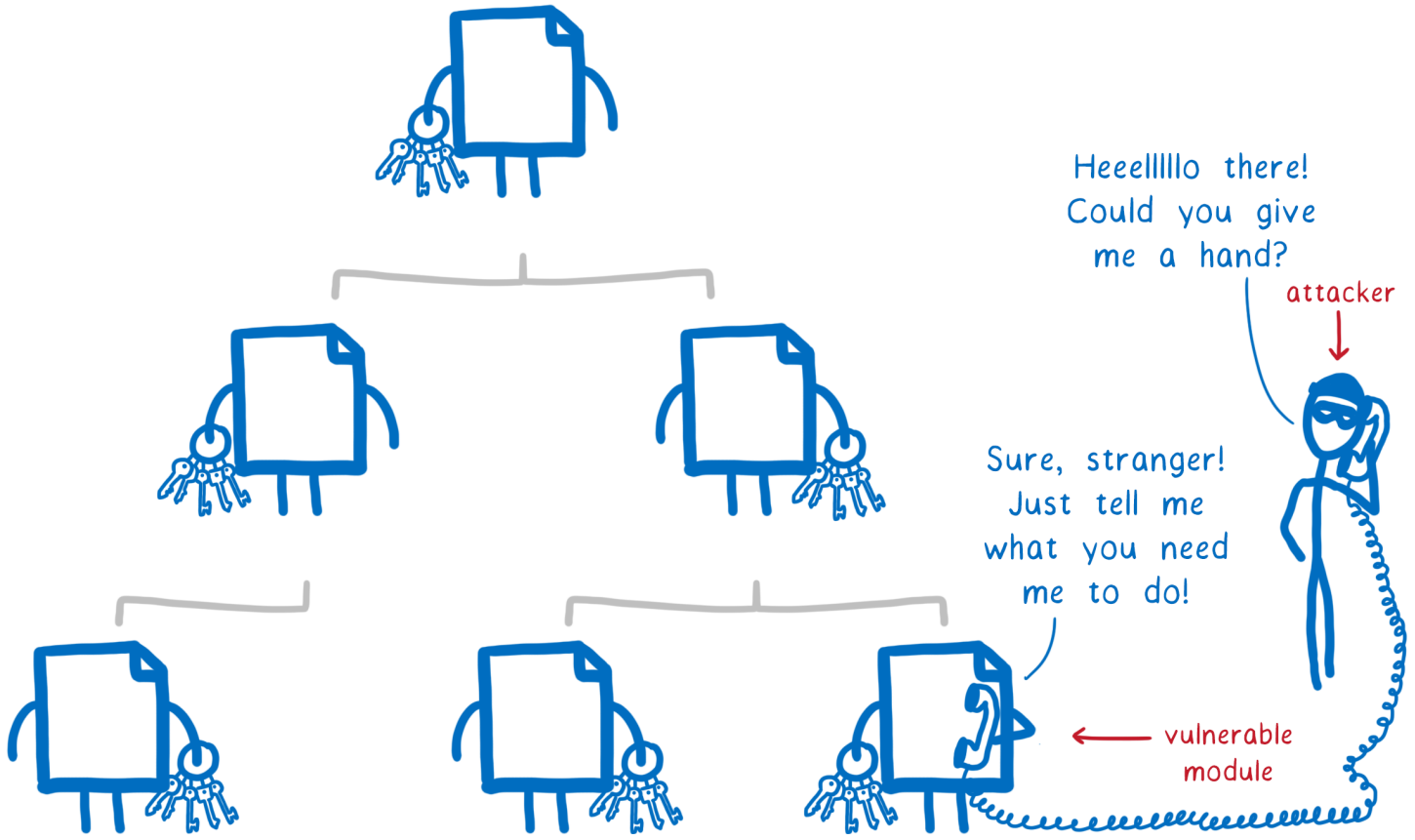
update

Malicious module



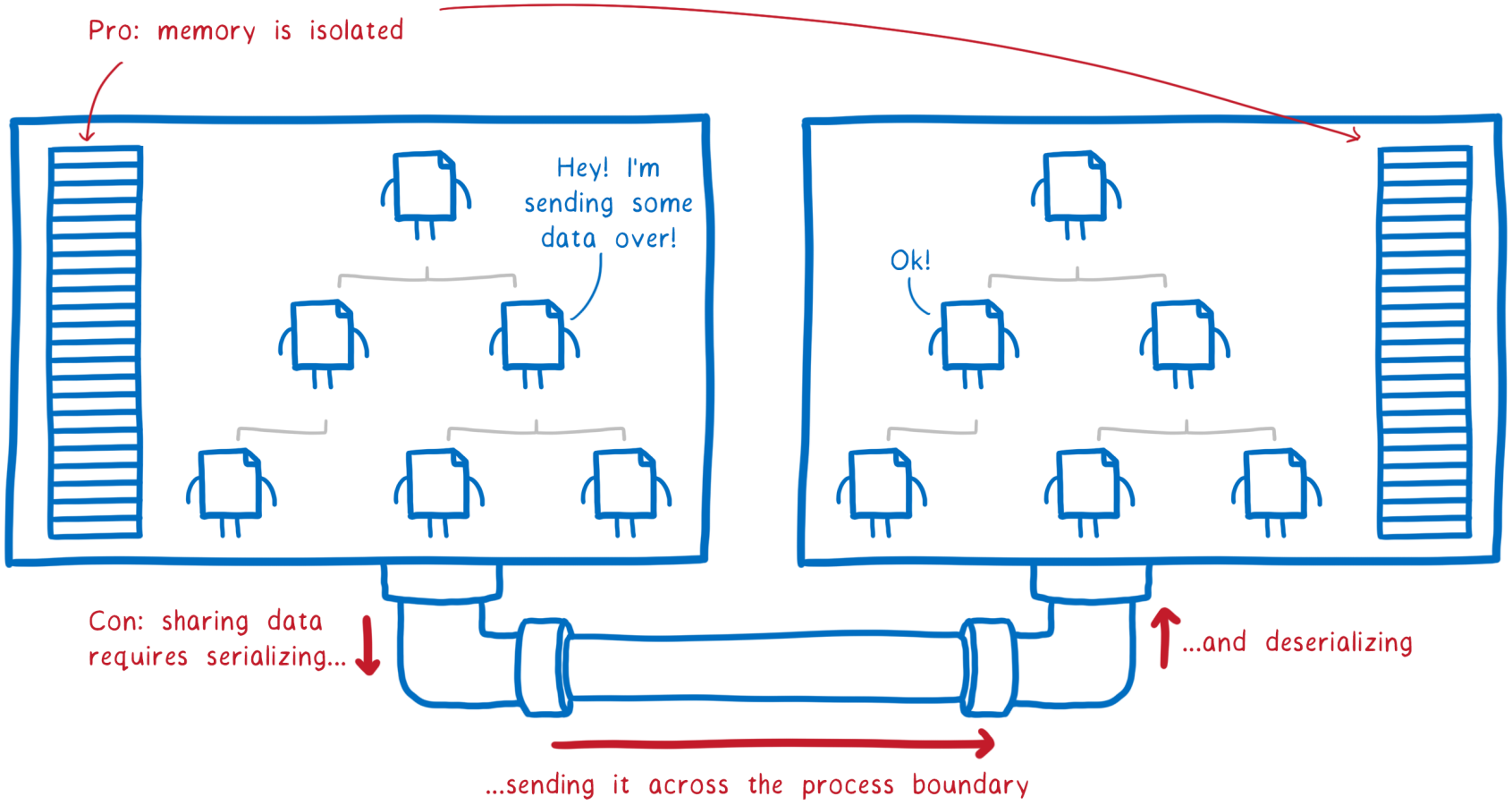
update

Vulnerable module



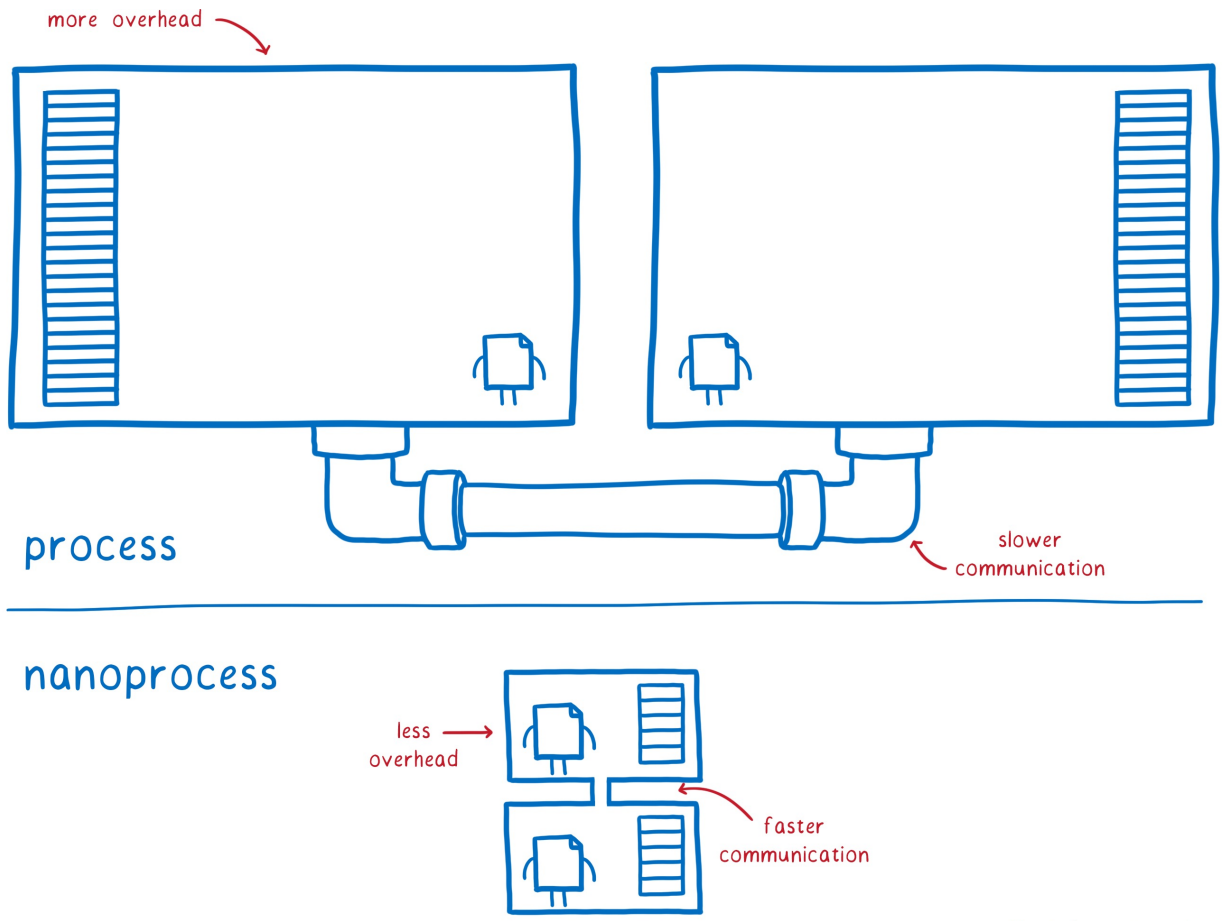
update

Process Isolation



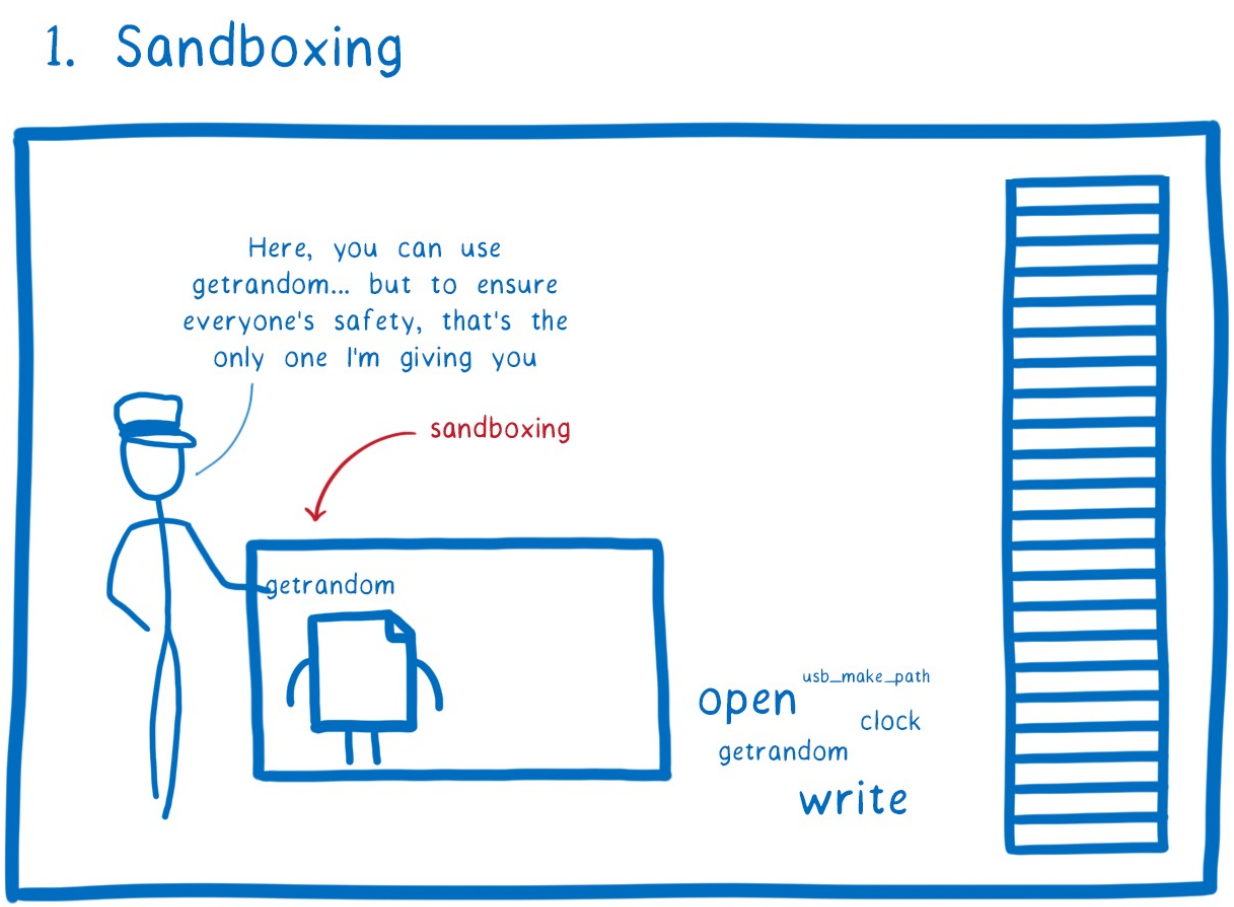
update

WebAssembly Nano-Process



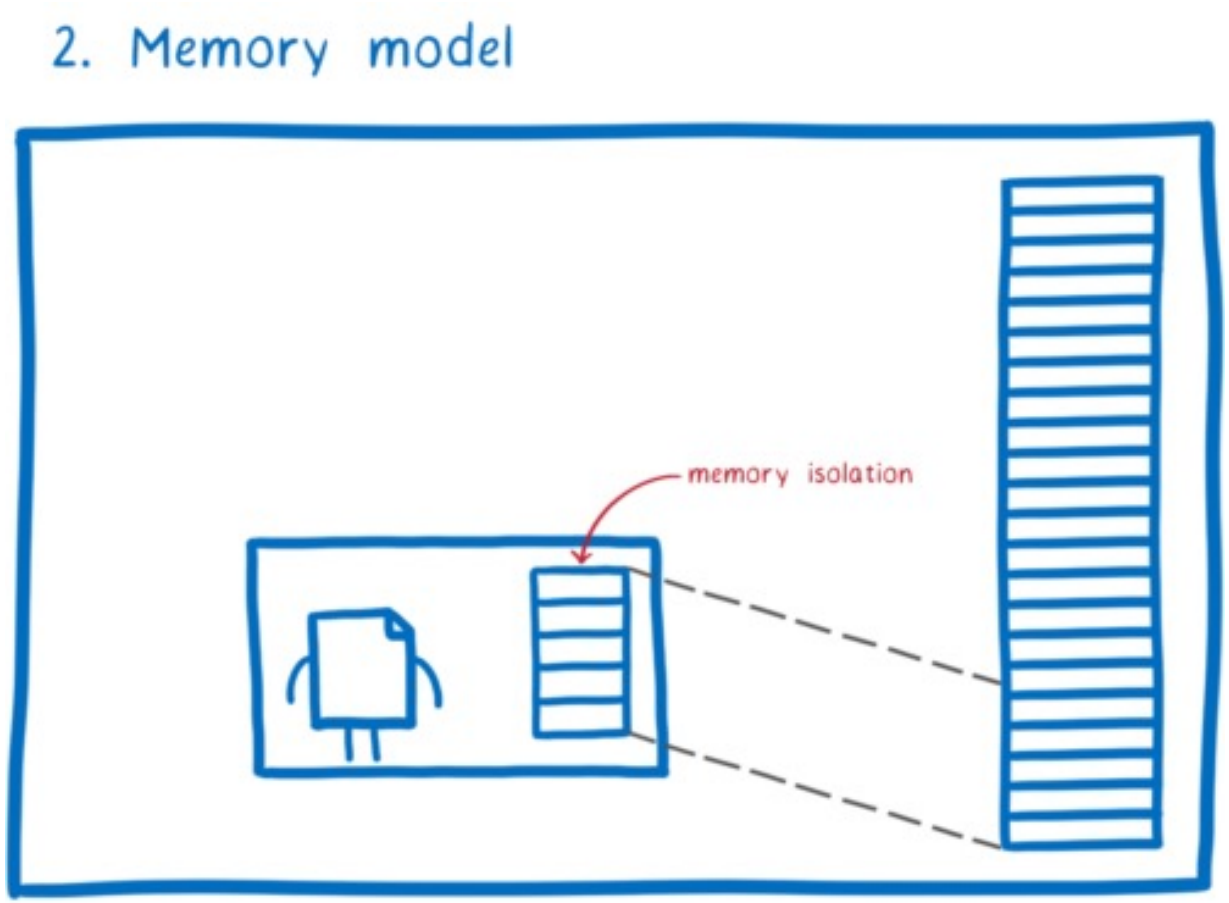
update

WebAssembly Nano-Process



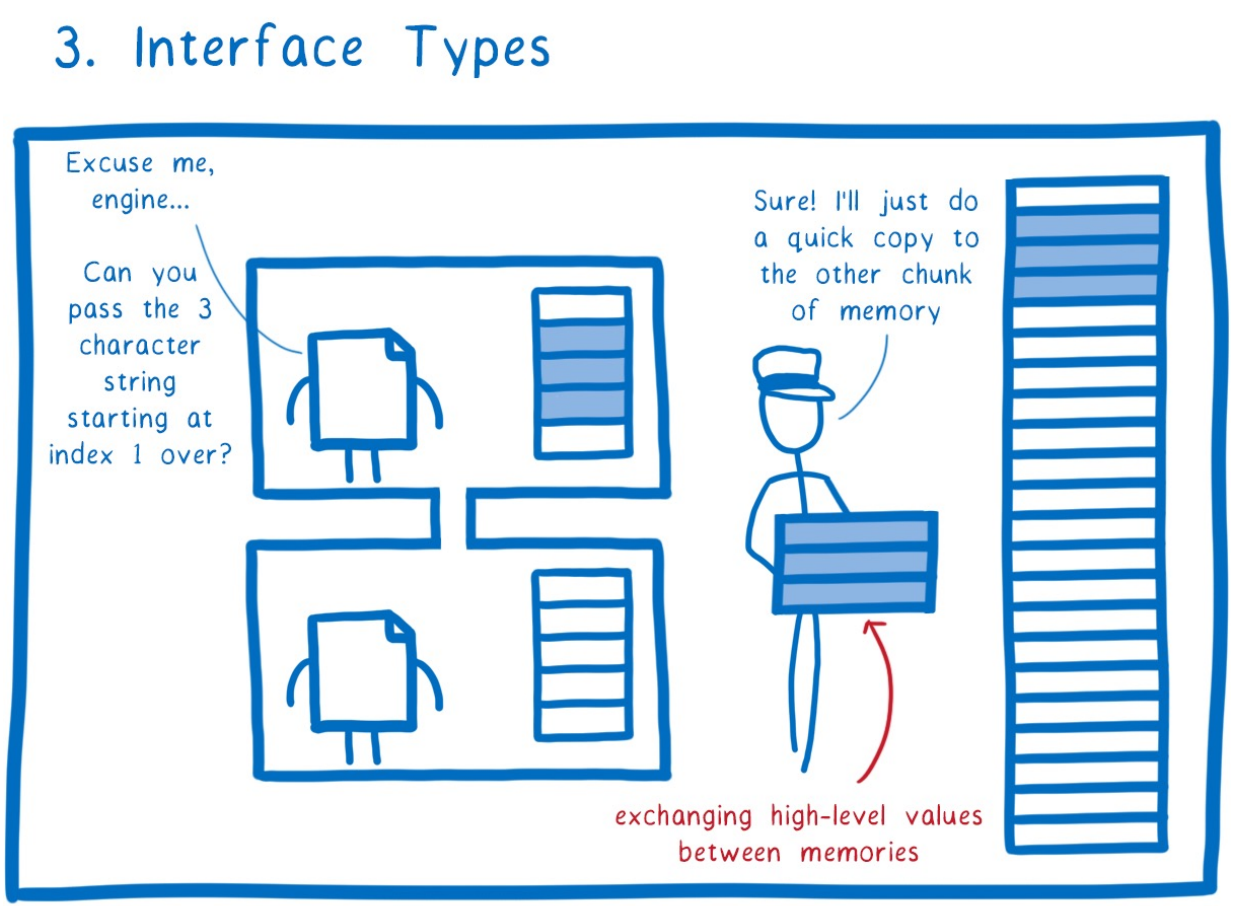
update

WebAssembly Nano-Process



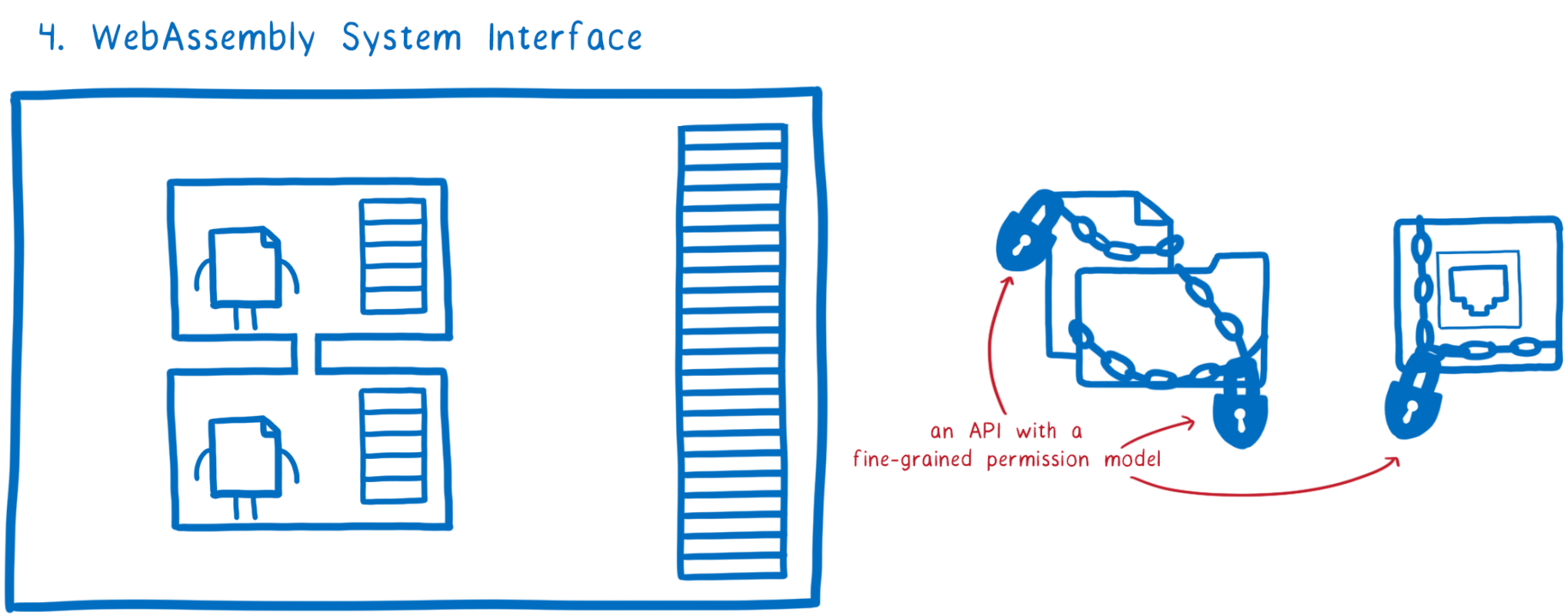
update

WebAssembly Nano-Process

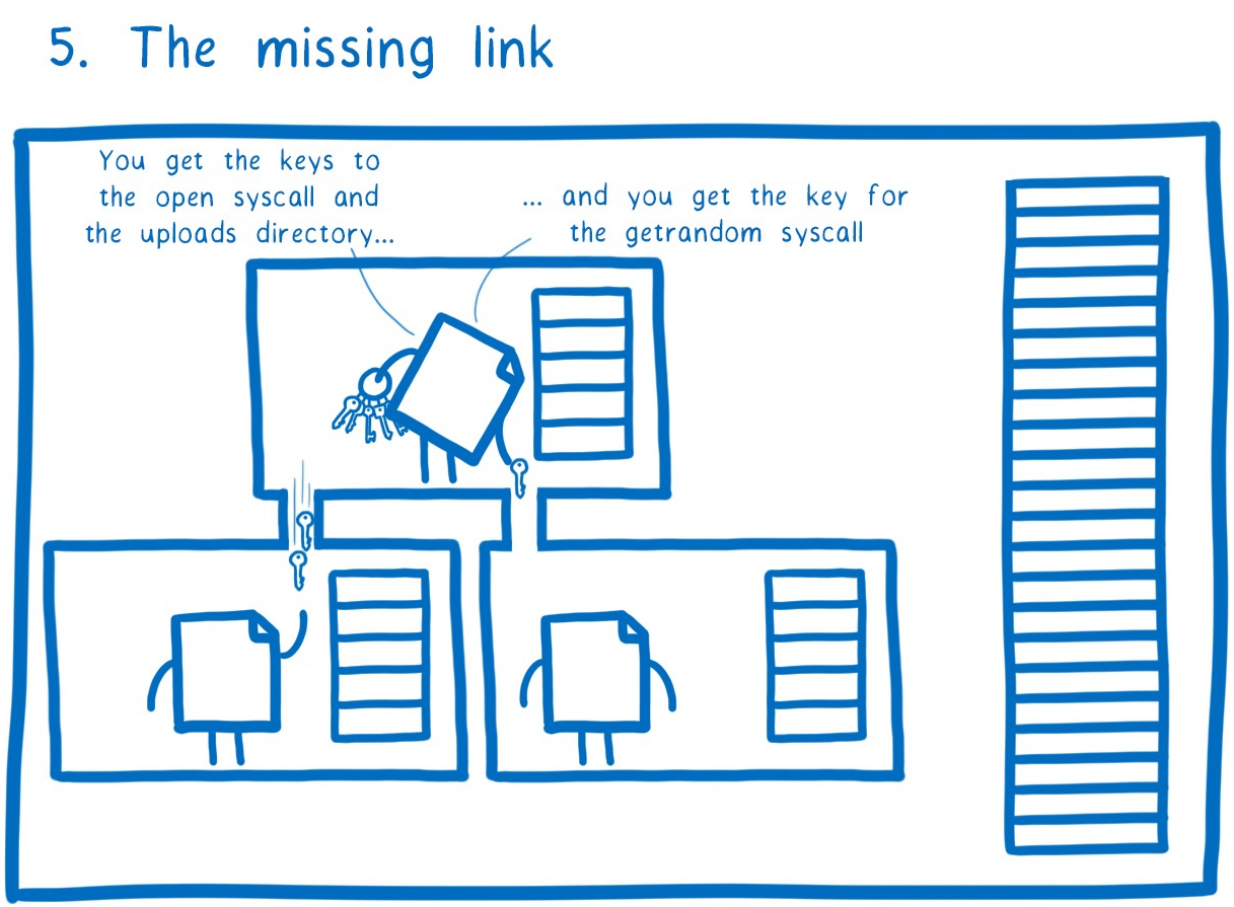


update

WebAssembly Nano-Process

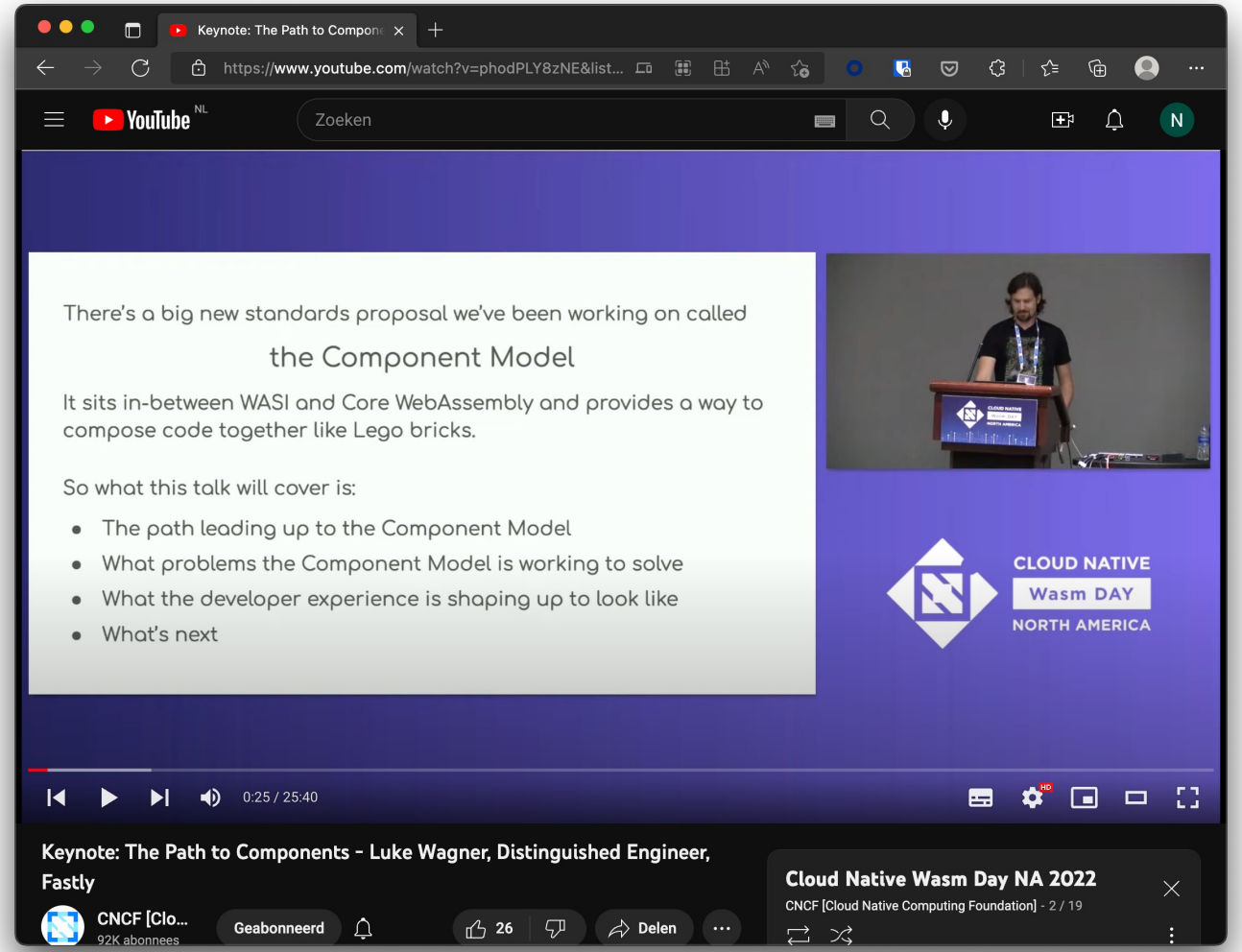


WebAssembly Nano-Process



update

WebAssembly Component Model



update

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

update

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

update

Questions?

- https://github.com/nielstanis/updateconference2022
- ntanis at Veracode.com
- https://blog.fennec.dev
- Děkuji! Thank you!

update