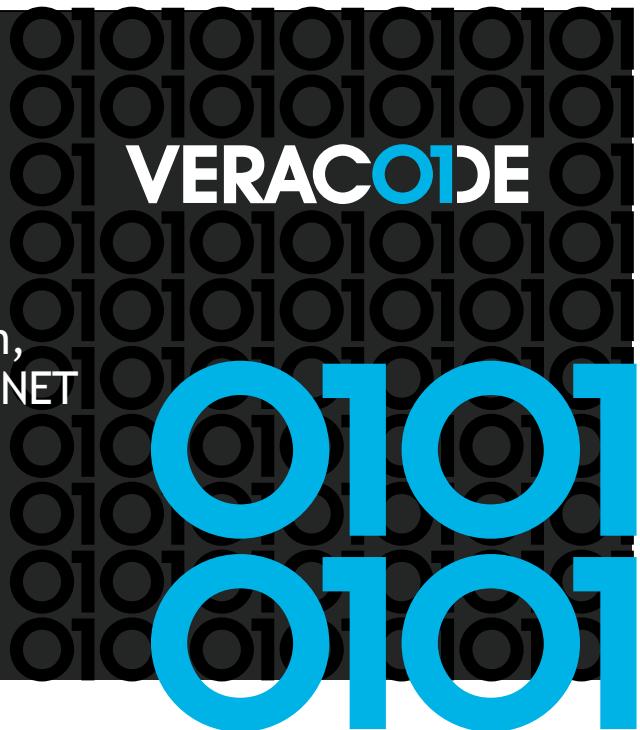




Using WebAssembly to run,
extend, and secure your .NET
application

Niels Tanis



0101
0101

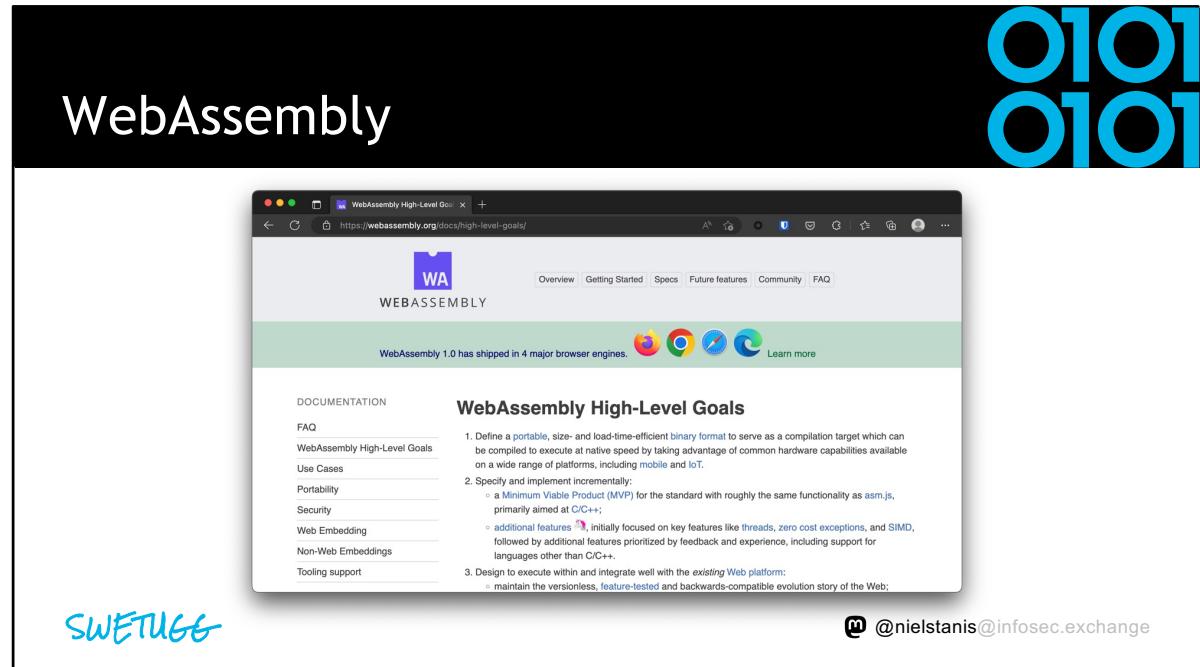
Who am I?

- 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



SWETUGG

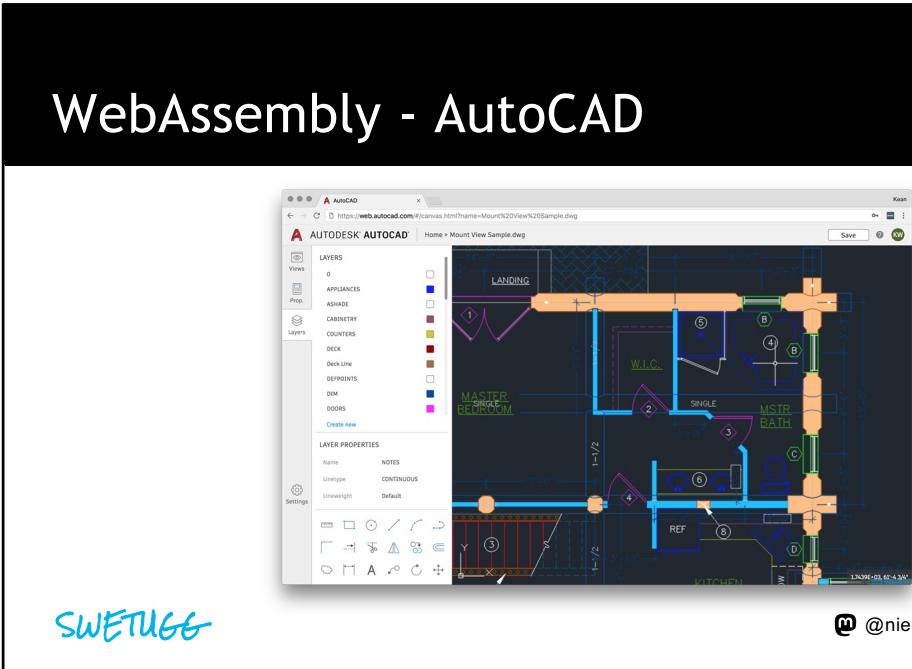
 @nielstanis@infosec.exchange



<https://hacks.mozilla.org/files/2019/08/04-01-star-diagram.png>

0101
0101

WebAssembly - AutoCAD



SWETUGG

0101
0101

WebAssembly - SDK's

A screenshot of a Medium article page. The title is "Introducing the Disney+ Application Development Kit (ADK)". It features a profile picture of Mike Hanley, a bio, and a summary of the post. The URL is <https://medium.com/disney-streaming/introducing-the-disney-application-development-kit-adk-ad85ca139073>.

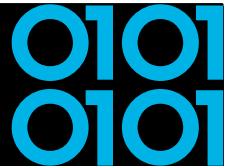
A screenshot of an Amazon Science blog post. The title is "How Prime Video updates its app for more than 8,000 device types". It includes a bio for the author, Alexandru Enă, and a summary of the post. The URL is <https://www.amazon.science/blog/how-prime-video-updates-its-app-for-more-than-8-000-device-types>.

SWETUGG

@nielstanis@infosec.exchange

<https://medium.com/disney-streaming/introducing-the-disney-application-development-kit-adk-ad85ca139073>

<https://www.amazon.science/blog/how-prime-video-updates-its-app-for-more-than-8-000-device-types>



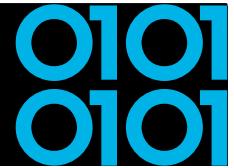
Agenda

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

SWETUGG

@nielstanis@infosec.exchange

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

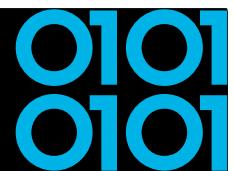


WEBASSEMBLY

SWETUGG

@nielstanis@infosec.exchange

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

SWETUGG

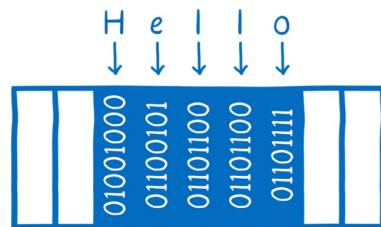
@nielstanis@infosec.exchange

<https://hacks.mozilla.org/2017/02/creating-and-working-with-webassembly-modules/>
<https://webassembly.org/>

0101
0101

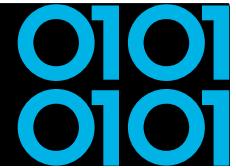
WebAssembly Memory

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



SWETUGG

@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!");
```

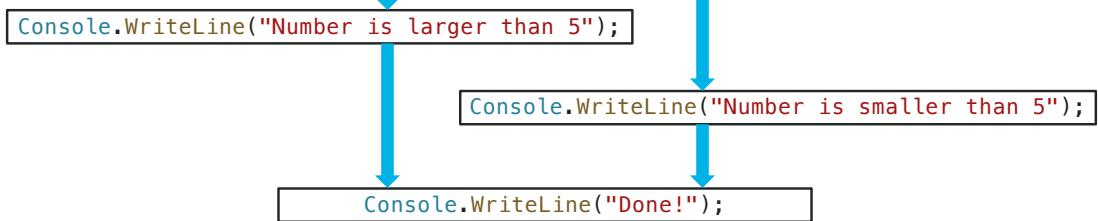
SWETUGG

 @nielstanis@infosec.exchange

0101
0101

WebAssembly Control-Flow Integrity

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



SWETUGG

@nielstanis@infosec.exchange

The screenshot shows a Firefox browser window displaying the RLBox website at <https://rlbox.dev>. The title bar says "Overview - Practical third-party library sandboxing with RLBox". The main content area features a logo of a book inside a red box with the word "RLBox" next to it. Below the logo, the word "Overview" is centered. The text explains that RLBox is a toolkit for sandboxing third-party C libraries, originally developed for Firefox and shipped with it since 2020. It describes the toolkit as consisting of a C++ framework and a Wasm backend. A link at the bottom leads to "rlbox.dev/chapters/rbbox-install.html". In the top right corner of the slide, there is a blue "0101" logo.

FireFox RLBox

SWETUGG

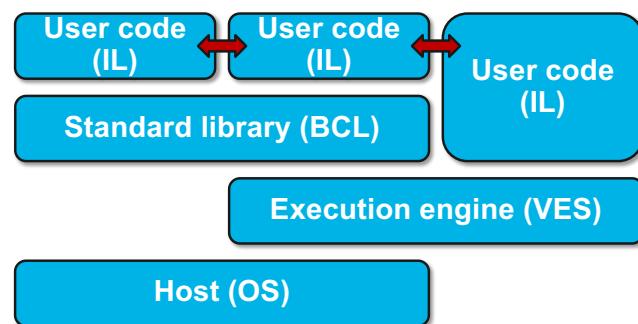
@nielstanis@infosec.exchange

<https://rlbox.dev/>

<https://hacks.mozilla.org/2020/02/securing-firefox-with-webassembly/>

Running .NET on WebAssembly

0101
0101



SWETUGG

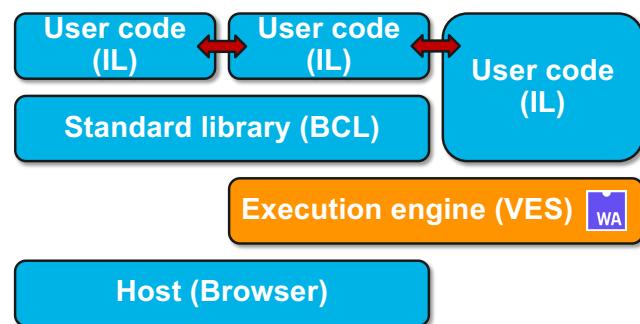
👤 @nielstanis@infosec.exchange

Diagram:

<https://github.com/itowlson/wasmday22/blob/main/slides/Wasm%20Interfaces%20and%20.NET.pptx>

Running .NET on WebAssembly

0101
0101



SWETUGG

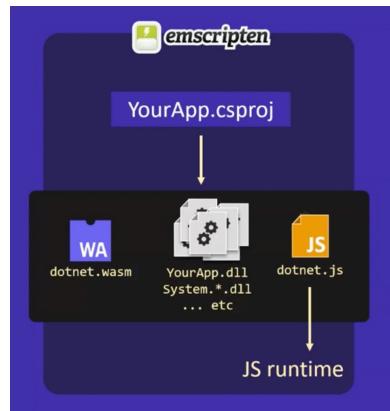
👤 @nielstanis@infosec.exchange

Diagram:

<https://github.com/itowlson/wasmday22/blob/main/slides/Wasm%20Interfaces%20and%20.NET.pptx>

0101
0101

Blazor WebAssembly

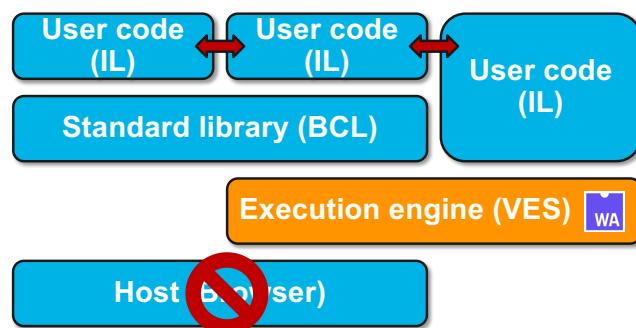


SWETUGG

✉️ @nielstanis@infosec.exchange

Running .NET on WebAssembly

0101
0101



SWETUGG

@nielstanis@infosec.exchange

Diagram:

<https://github.com/itowlson/wasmday22/blob/main/slides/Wasm%20Interfaces%20and%20.NET.pptx>

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

SWETUGG

 @nielstanis@infosec.exchange

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? 😊

SWETUGG

 @nielstanis@infosec.exchange

0101
0101

Docker vs WASM & WASI

The screenshot shows a dark-themed Twitter interface. On the left, there's a sidebar with various icons for navigation. The main content area displays a tweet from user @solomonstrel. The tweet reads:

If WASM+WASI existed in 2008, we wouldn't have needed to created Docker. That's how important it is. Webassembly on the server is the future of computing. A standardized system interface was the missing link. Let's hope WASI is up to the task!

Below this tweet, there is a reply from user @linclark. The reply reads:

WebAssembly running outside the web has a huge future. And that future gets one giant leap closer today with...

Announcing WASI: A system interface for running WebAssembly outside the web (and inside it too)

hacks.mozilla.org/2019/03/standards/

Dit bericht toont collectie weergeven

At the bottom right of the screenshot, there is a watermark or signature that says "@nielstanis@infosec.exchange".

SWETUGG

0101
0101

Docker vs WASM & WASI



@nielstanis@infosec.exchange

SWETUGG

0101
0101

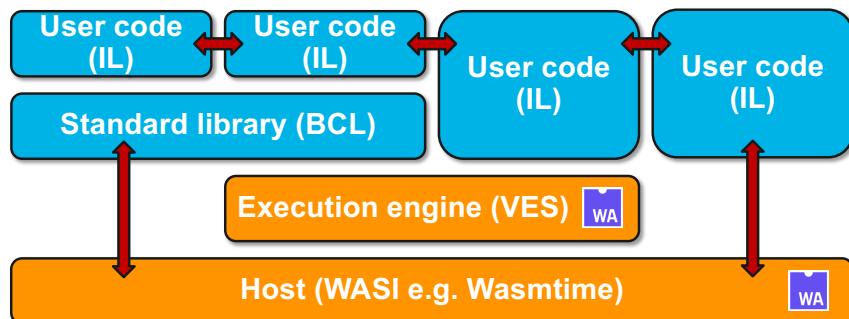
Docker & WASM

The screenshot displays two browser windows side-by-side. The left window shows the official Docker+Wasm Technical Preview landing page. It 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 background with white text. It features a large heading 'Introducing the Docker+Wasm Technical Preview' and a bio for 'MICHAEL IRWIN' with a small profile picture. Below the bio is the date 'Oct 24 2022'. A note at the bottom states: '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.' The right window shows a detailed architectural diagram of the Docker Engine. It starts with a 'Docker Engine' box at the top, which connects to a 'containerd' box. Inside the 'containerd' box, there are three separate boxes labeled 'containerd-shim', each containing a 'runc' box and a 'Container process' box. To the right of these is another box labeled 'containerd-wasm-shim', which contains a 'wasmedge' box and a 'Wasm Module' box.

SWETUGG

@nielstanis@infosec.exchange

WebAssembly System Interface WASI

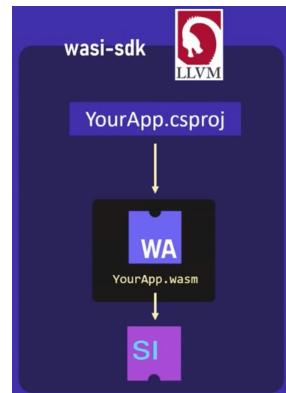


SWETUGG

@nielstanis@infosec.exchange

0101
0101

Experimental WASI SDK for .NET



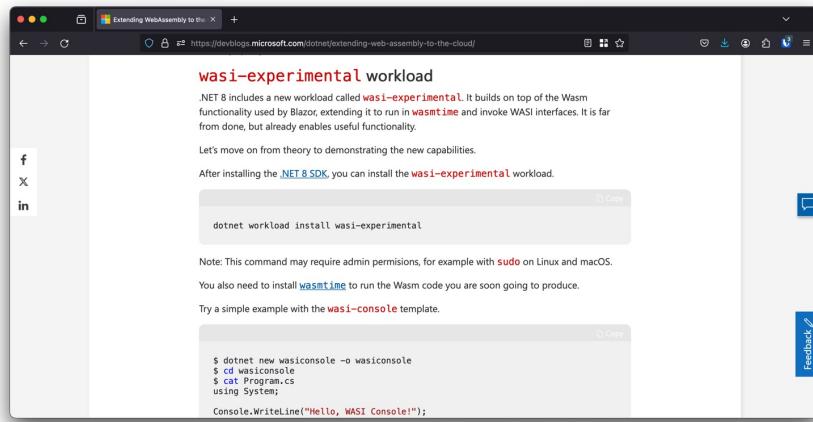
SWETUGG

@nielstanis@infosec.exchange

<https://github.com/SteveSandersonMS/dotnet-wasi-sdk>

.NET 8 WASI-Experimental

0101
0101



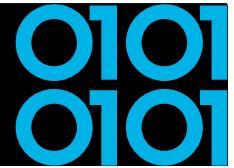
SWETUGG

@nielstanis@infosec.exchange

<https://github.com/dotnet/runtime/issues/65895>

<https://github.com/SteveSandersonMS/dotnet-wasi-sdk>

<https://devblogs.microsoft.com/dotnet/extending-web-assembly-to-the-cloud/>



Extending .NET with WASM

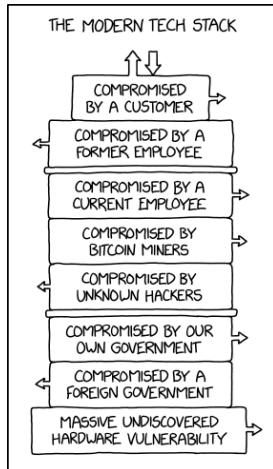
- WasmTime.NET NuGet package
- Can run WASM inside of any .NET application
- Extend with Rust based WASM module
- Limit capabilities
- Demo time!

SWETUGG

 @nielstanis@infosec.exchange

0101
0101

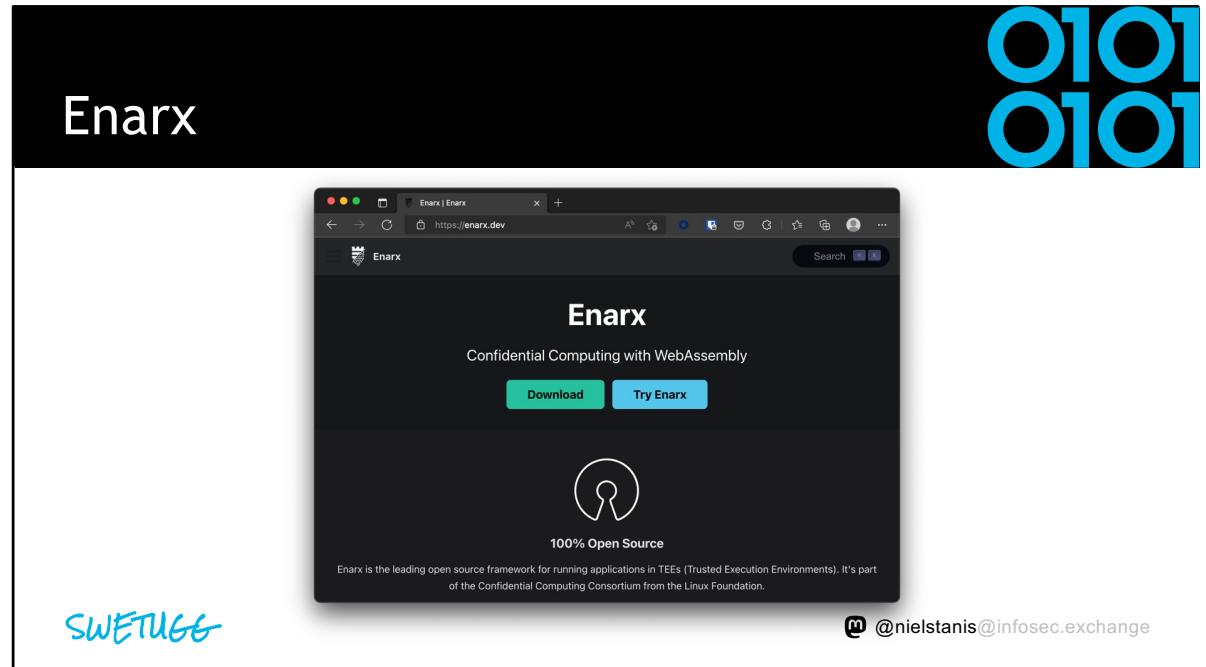
Trusted Computing - XKCD 2166



SWETUGG

@nielstanis@infosec.exchange

<https://xkcd.com/2166/>



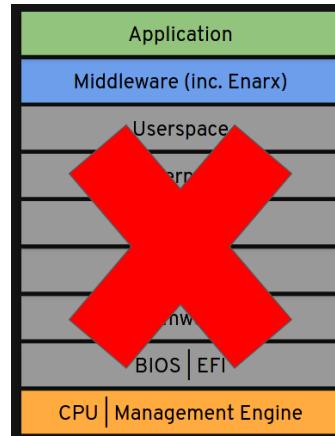
<https://enarx.dev/>

0101
0101

Enarx Threat Model

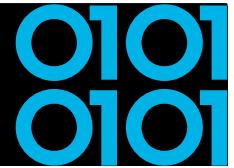
- Don't trust the host
- Don't trust the host owner
- Don't trust the host operator
- Hardware cryptographically verified
- Software audited and cryptographically verified

SWETUGG

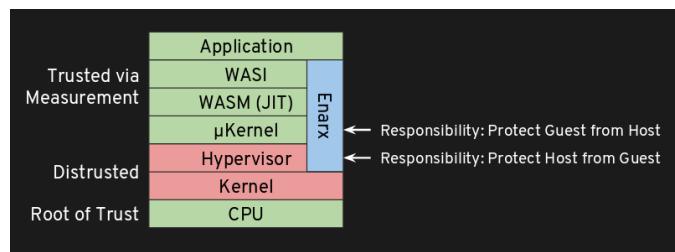


https://enarx.dev/doc/threat-model.html

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

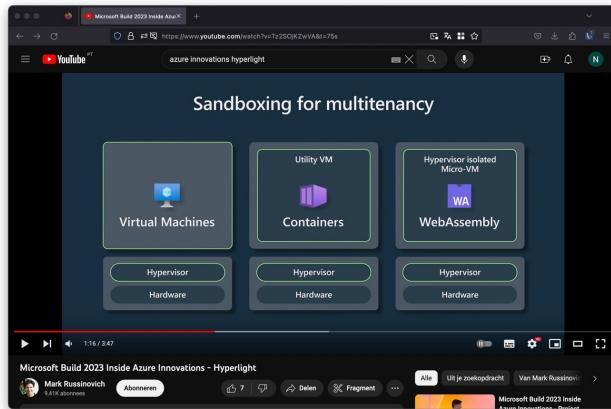


SWETUGG

@nielstanis@infosec.exchange

0101
0101

Project Hyperlight

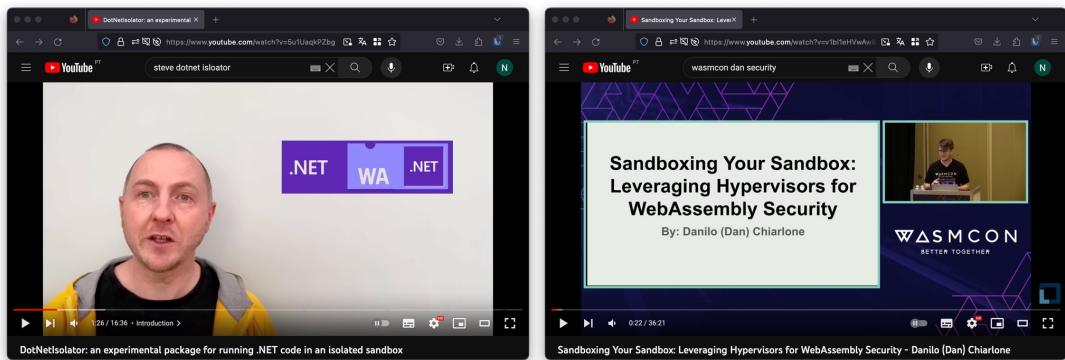


SWETUGG

@nielstanis@infosec.exchange

DotNetIsolator & Project Hyperlight

0101
0101



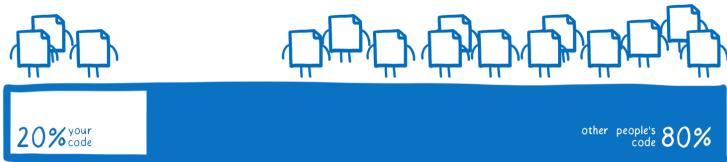
SWETUGG

@nielstanis@infosec.exchange

0101
0101

WASM - What's next?

composition of an
average code base



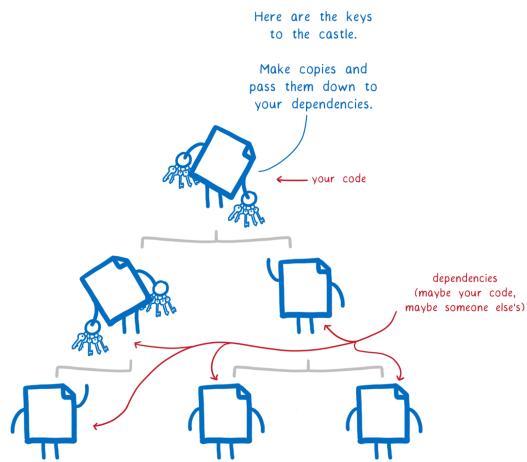
SWETUGG

 @nielstanis@infosec.exchange

0101
0101

Dependencies

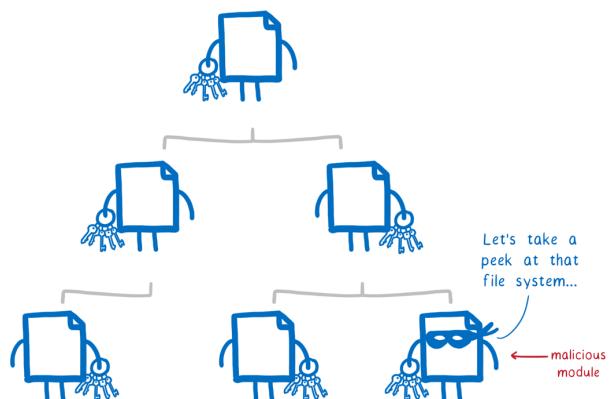
SWETUGG



iis@infosec.exchange

0101
0101

Malicious module

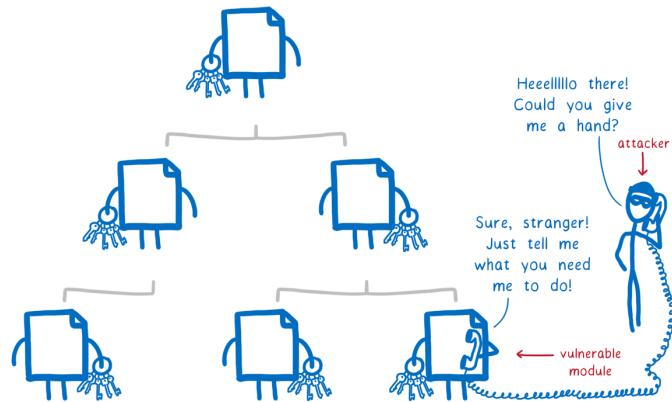


SWETUGG

@nielstanis@infosec.exchange

0101
0101

Vulnerable module

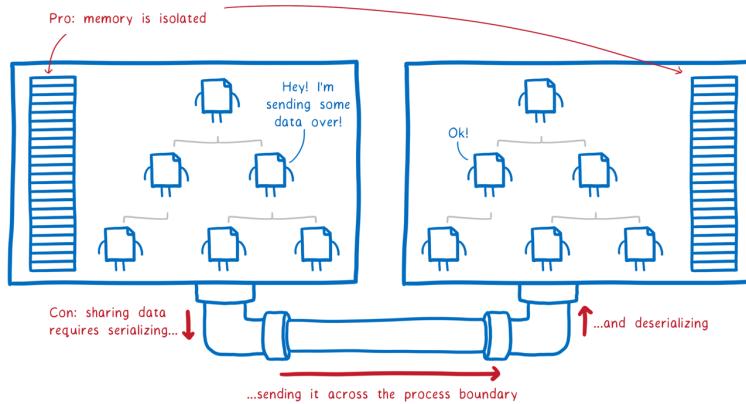


SWETUGG

@nielstanis@infosec.exchange

0101
0101

Process Isolation

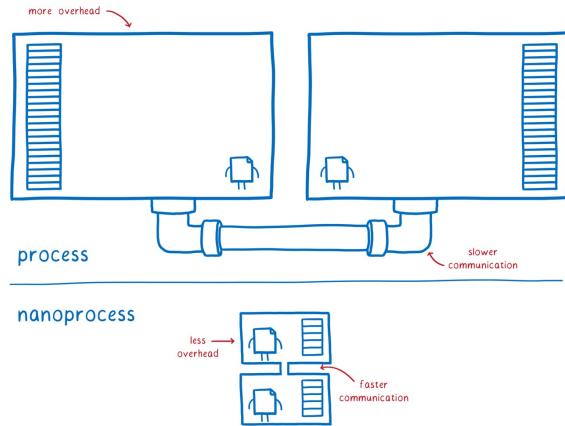


SWETUGG

@nielstanis@infosec.exchange

0101
0101

WebAssembly Nano-Process



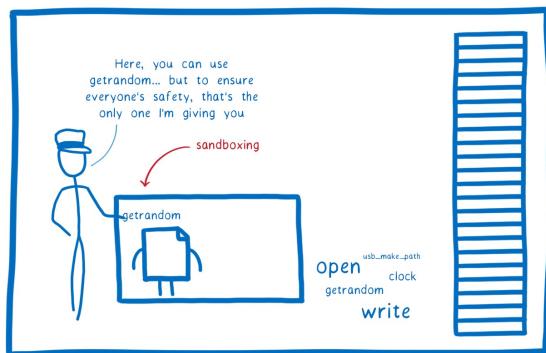
* not drawn to scale
m @nielstanis@infosec.exchange

SWETUGG

0101
0101

WebAssembly Nano-Process

1. Sandboxing



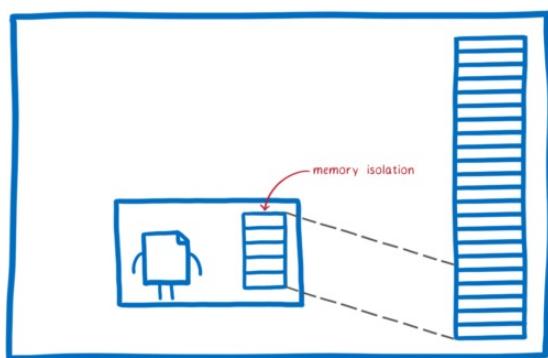
SWETUGG

 @nielstanis@infosec.exchange

0101
0101

WebAssembly Nano-Process

2. Memory model



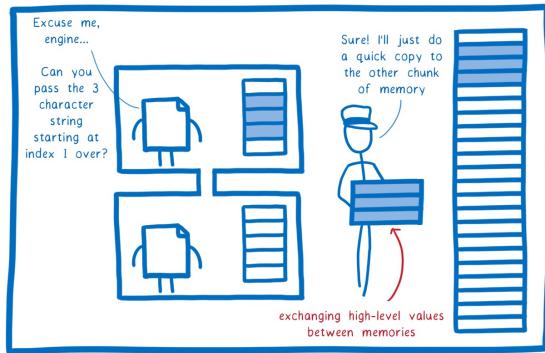
@nielstanis@infosec.exchange

SWETUGG

0101
0101

WebAssembly Nano-Process

3. Interface Types



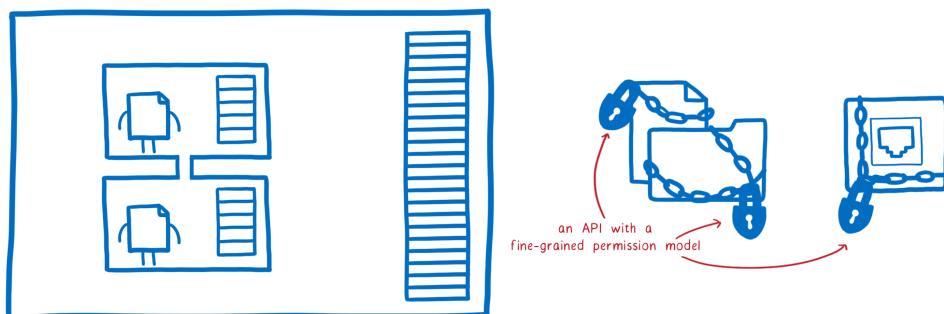
@nielstanis@infosec.exchange

SWETUGG

0101
0101

WebAssembly Nano-Process

4. WebAssembly System Interface



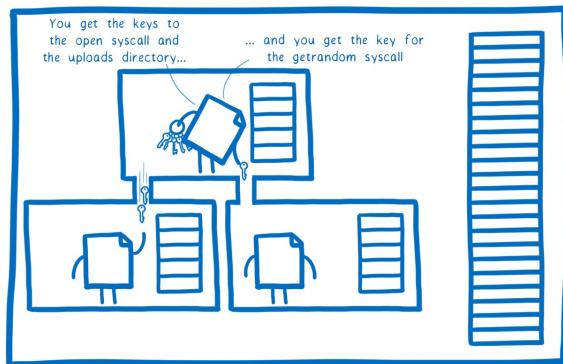
SWETUGG

@nielstanis@infosec.exchange

0101
0101

WebAssembly Nano-Process

5. The missing link



@nielstanis@infosec.exchange

SWETUGG

0101
0101

WebAssembly Component Model

The screenshot shows a YouTube video player window. The video title is "Keynote: What is a Component? (and Why)? - Luke Wagner, Distinguished Engineer, Fasty". The video is 29:50 minutes long, currently at 0:57. The video content shows a man speaking on stage at a conference booth for "WASMCON BETTER TOGETHER". The video player interface includes standard controls like play/pause, volume, and a progress bar.

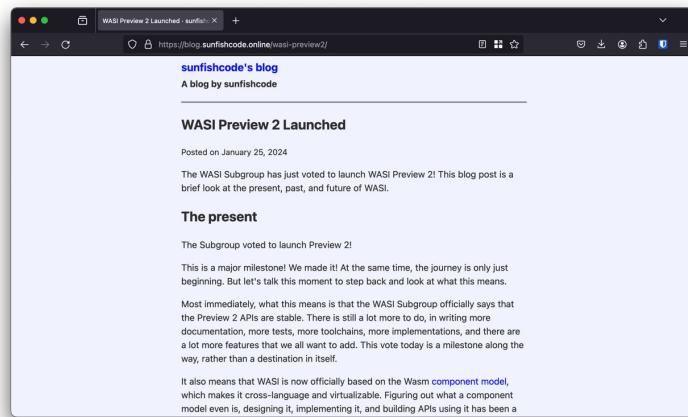
SWETUGG

@nielstanis@infosec.exchange

<https://www.youtube.com/watch?v=tAACYA1Mwv4>

WASI Preview 2

0101
0101

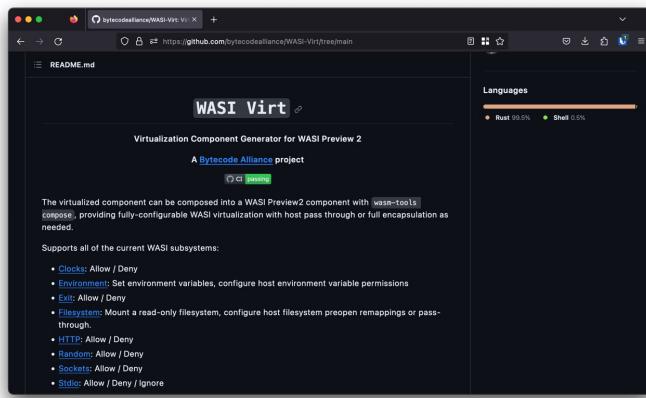


SWETUGG

@nielstanis@infosec.exchange

0101
0101

WASI Virt



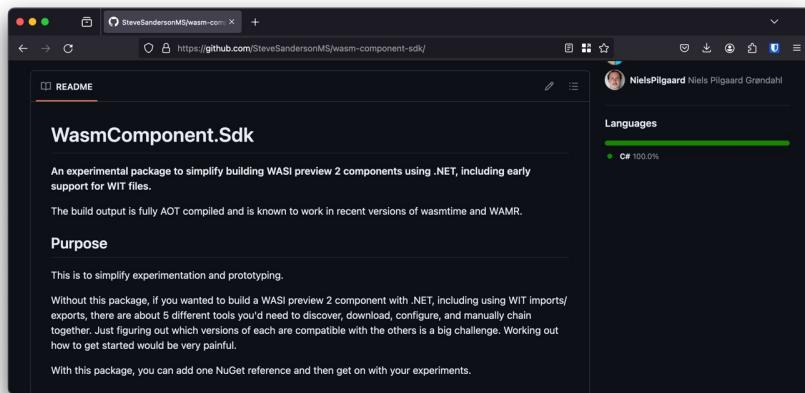
SWETUGG

@nielstanis@infosec.exchange

<https://www.youtube.com/watch?v=tAACYA1Mwv4>

WasmComponent.SDK

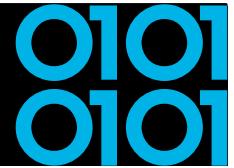
0101
0101



SWETUGG

@nielstanis@infosec.exchange

<https://github.com/SteveSandersonMS/wasm-component-sdk/>



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

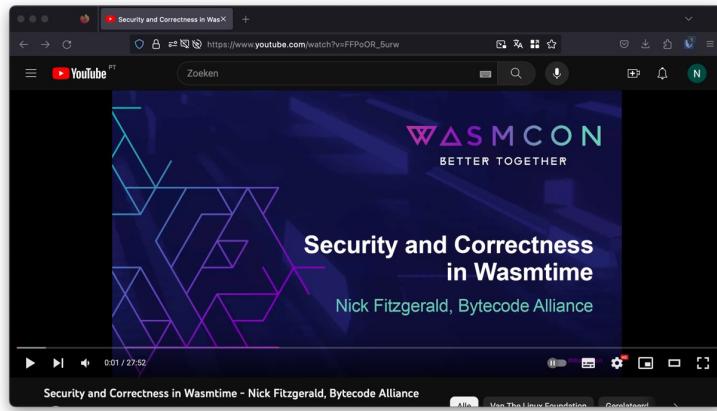
SWETUGG

@nielstanis@infosec.exchange

<https://bytecodealliance.org/articles/security-and-correctness-in-wasmtime>

0101
0101

Runtimes and Security



SWETUGG

@nielstanis@infosec.exchange

https://www.youtube.com/watch?v=FFPoOR_5urw

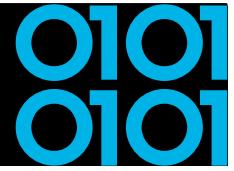


Conclusion

- 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!

SWETUGG

@nielstanis@infosec.exchange



Questions?

- <https://github.com/nielstanis/swetugg2024>
- ntanis at Veracode.com
- @nielstanis@infosec.exchange
- <https://blog.fennec.dev>
- Tack! Thank you!

SWETUGG

 [@nielstanis@infosec.exchange](mailto:nielstanis@infosec.exchange)