CYBER ON BOARD

Wasefire

Julien Cretin, Jean-Michel Picod
– Google

CyberOnBoard 2025

Firmware security issues are still too frequent

CVE List▾ CNAs▾ WGs▾ News & Blog▾ Board▾ About▾ NVD

Go to for: CVSS Scores CPE Info

Search CVE List | Downloads | Data Feeds | Update a CVE Record | Request CVE IDs

TOTAL CVE Records: 215058

NOTICE: Transition to the all-new CVE website at WWW.CVE.ORG and CVE Record Format JSON are underway.

NOTICE: Legacy CVE List download formats will be phased out beginning January 1, 2024.
New CVE List download format is available now.

HOME > CVE > SEARCH RESULTS

Search Results

There are 4215 CVE Records that match your search.

Name	Description
CVE-2023-5746	A vulnerability regarding use of externally-controlled format string is found in the cgi component. This allows remote attack following models with Synology Camera Firmware versions before 1.0.5-0185 may be affected: BC500 and TC500.
CVE-2023-5409	HP is aware of a potential security vulnerability in HP t430 and t638 Thin Client PCs. These models may be susceptible to a system firmware using a publicly disclosed private key. HP is providing recommended guidance for customers to reduce exp
CVE-2023-5288	A remote unauthorized attacker may connect to the SIM1012, interact with the device and change configuration settings. T a new firmware version to the device.
CVE-2023-4929	All firmware versions of the NPort 5000 Series are affected by an improper validation of integrity check vulnerability. This v upgrades, potentially allowing malicious users to manipulate the firmware and gain control of devices.
CVE-2023-4692	An out-of-bounds write flaw was found in grub2's NTFS filesystem driver. This issue may allow an attacker to present a spe metadata corruption. In some circumstances, the attack may also corrupt the UEFI firmware heap metadata. As a result, a achieved.
CVE-2023-45194	Use of default credentials vulnerability in MR-GM2 firmware Ver. 3.00.03 and earlier, and MR-GM3 (-D/-K/-S/-DK/-DKS/-M/ unauthenticated attacker to intercept wireless LAN communication, when the affected product performs the communicatior configuration.
CVE-2023-43627	Path traversal vulnerability in ACERA 13?0 firmware ver.01.26 and earlier, and ACERA 1310 firmware ver.01.26 and earlier information such as system files by senc
CVE-2023-43478	fake_upload.cgi on the Telstra Smart Mo backups, which could allow them to alte
CVE-2023-43477	The ping from parameter of ping tracer

Google Security Blog

The latest news and insights from Google on security and safety on the Internet

Advisory: Security Issue with Bluetooth Low Energy (BLE) Titan Security Keys

May 15, 2019

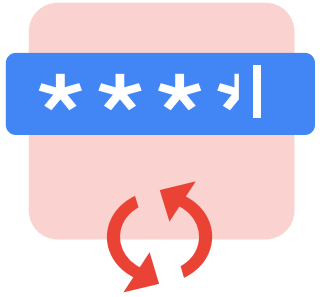
Posted by Christiaan Brand, Product Manager, Google Cloud

Google

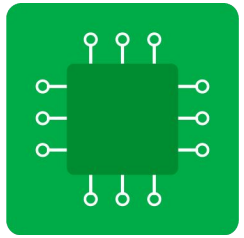
Security and Privacy Research

Common security issues

Memory corruption

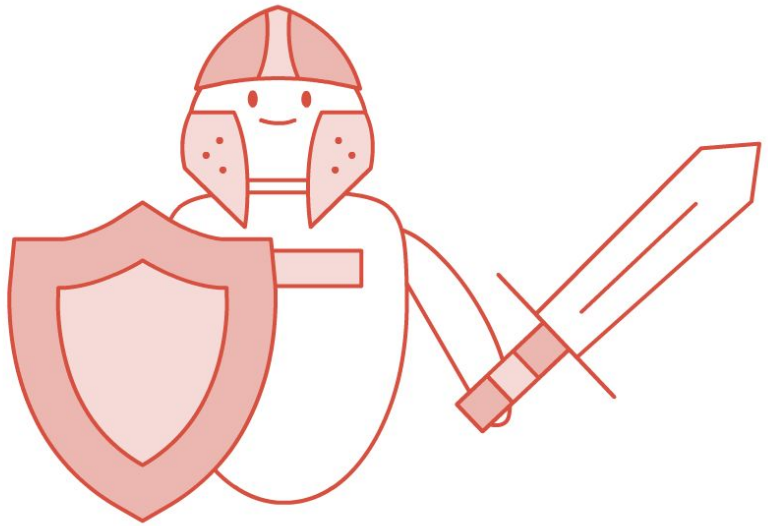


Default credentials



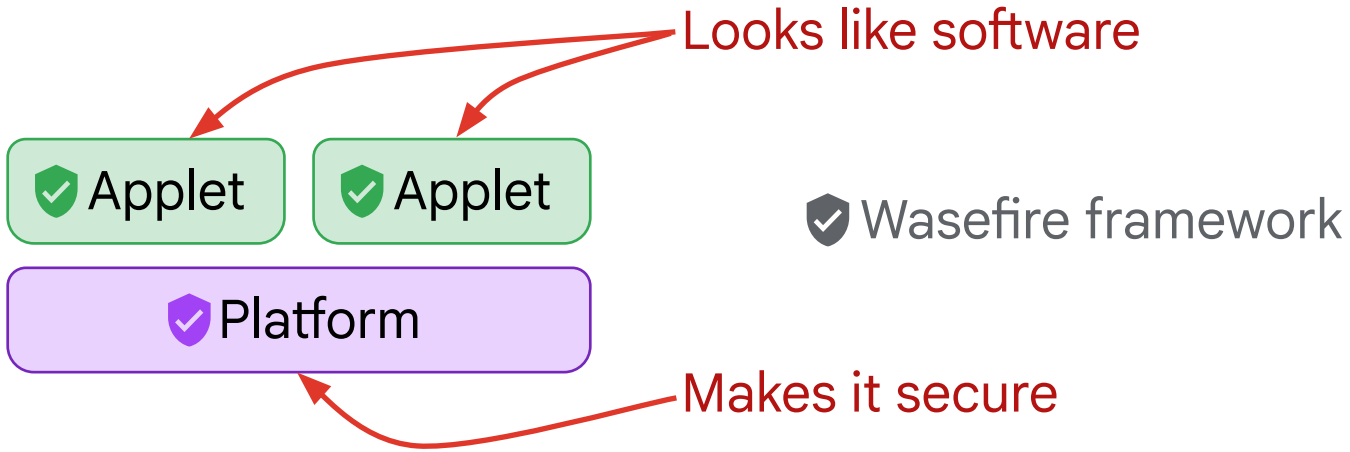
Side channel attacks

Manual or no updates



We need a secure platform that takes care of the security on behalf of the developer

Introducing Wasefire



Agenda

Goals and properties

Architecture and design

Existing solutions

Current state and future work

Wasefire for you

Goals and properties

Secure by default

Secure by default

Hardened platform

Built with side-channel attacks and fault injections in mind

Secure by default

Hardened platform

Built with side-channel attacks and fault injections in mind

Sandboxed applets

Applets need specific API permissions to interact with the platform (and hardware)

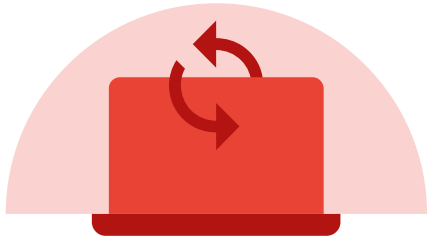
Secure by default

Hardened platform

Built with side-channel attacks and fault injections in mind

Sandboxed applets

Applets need specific API permissions to interact with the platform (and hardware)

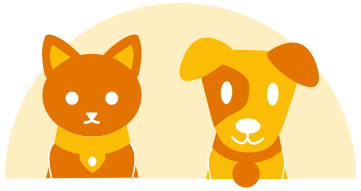


All upgradable

Platform and applets are upgradable by design

For all software engineers

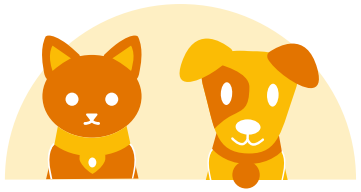
For all software engineers



Language agnostic

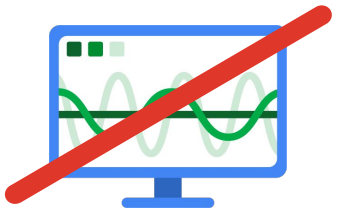
Developers can use the language, IDE, and workflow they are most comfortable with

For all software engineers



Language agnostic

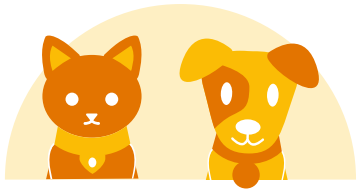
Developers can use the language, IDE, and workflow they are most comfortable with



Hardware independent

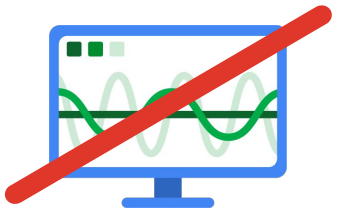
Development can be done on a desktop machine without special hardware

For all software engineers



Language agnostic

Developers can use the language, IDE, and workflow they are most comfortable with



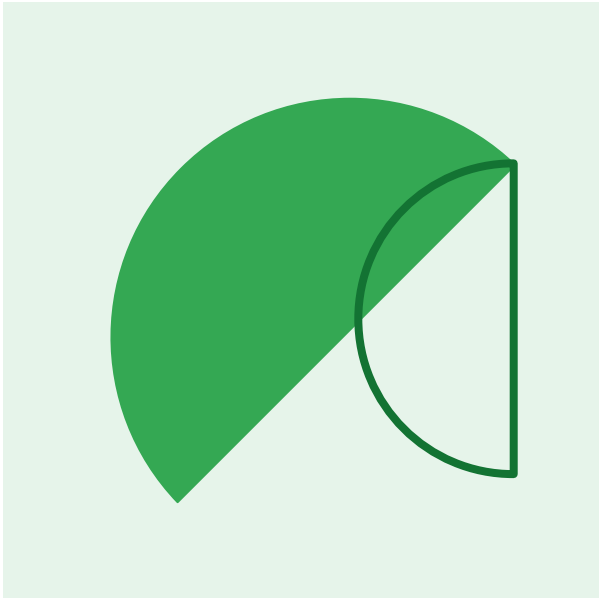
Hardware independent

Development can be done on a desktop machine without special hardware

Open source

Hardware vendors can provide proprietary platforms

Architecture and design



Google

Security and Privacy Research

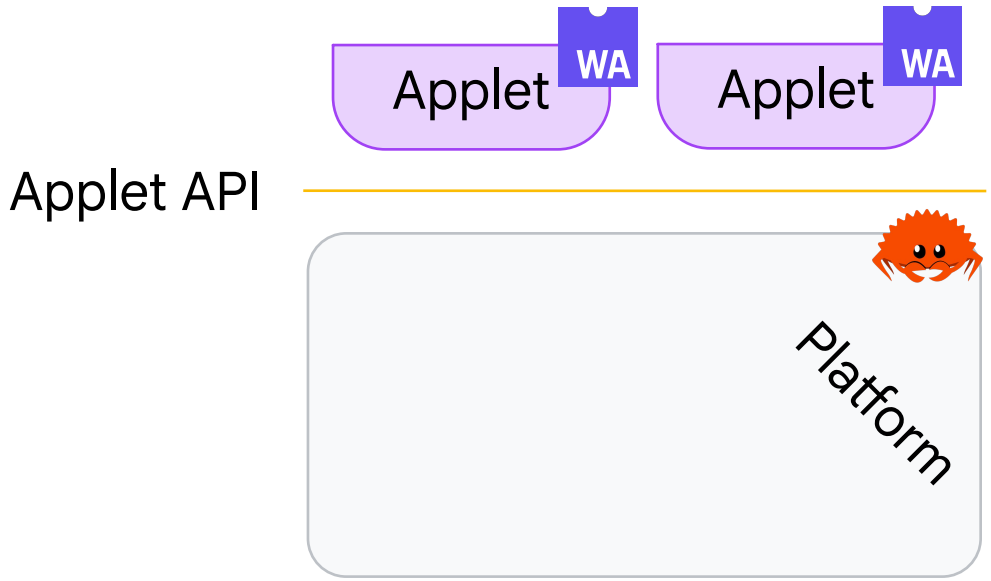
WebAssembly virtual machine

From the WebAssembly specification:

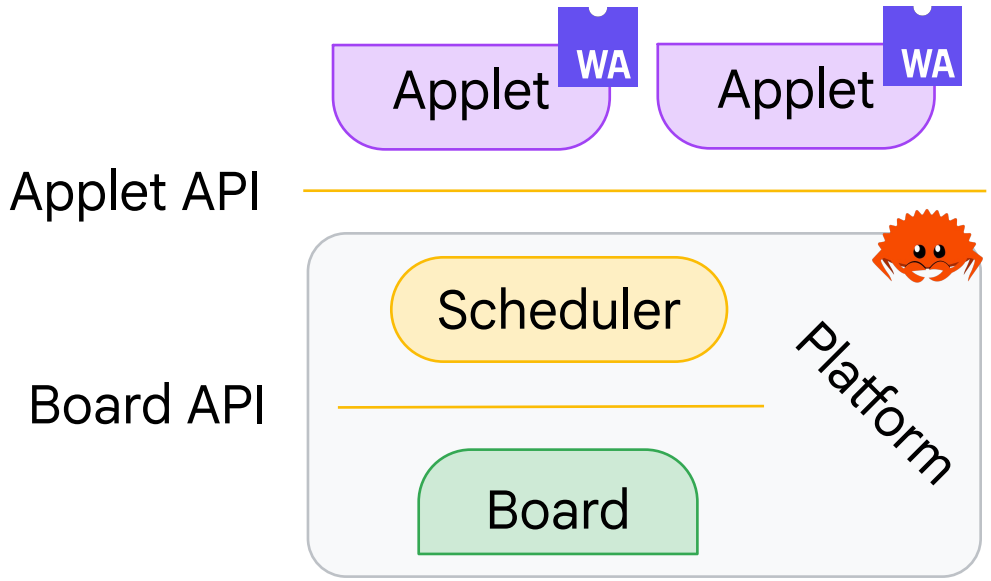
> WebAssembly (abbreviated Wasm [1]) is a *safe, portable, low-level code format* designed for efficient execution and compact representation. Its main goal is to enable high performance applications on the Web, but it does not make any Web-specific assumptions or provide Web-specific features, so it can be employed in other environments as well.

We use it in embedded environment for its safe (sandboxing), portable, and compact format.

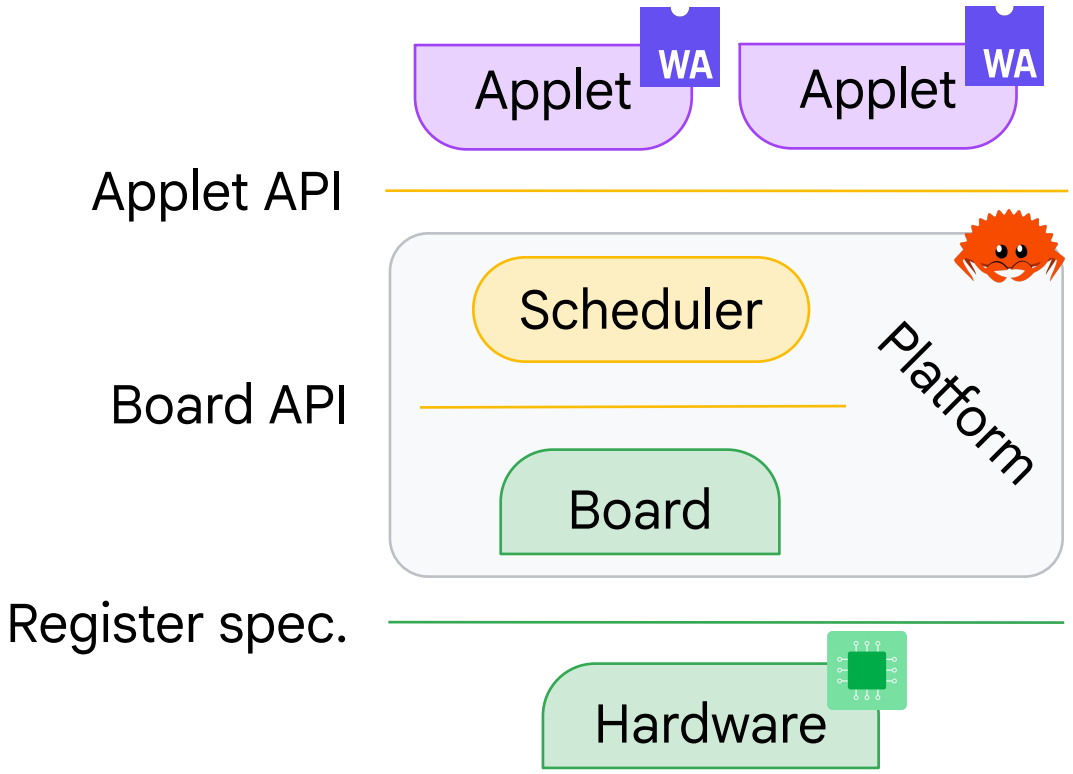
WebAssembly virtual machine



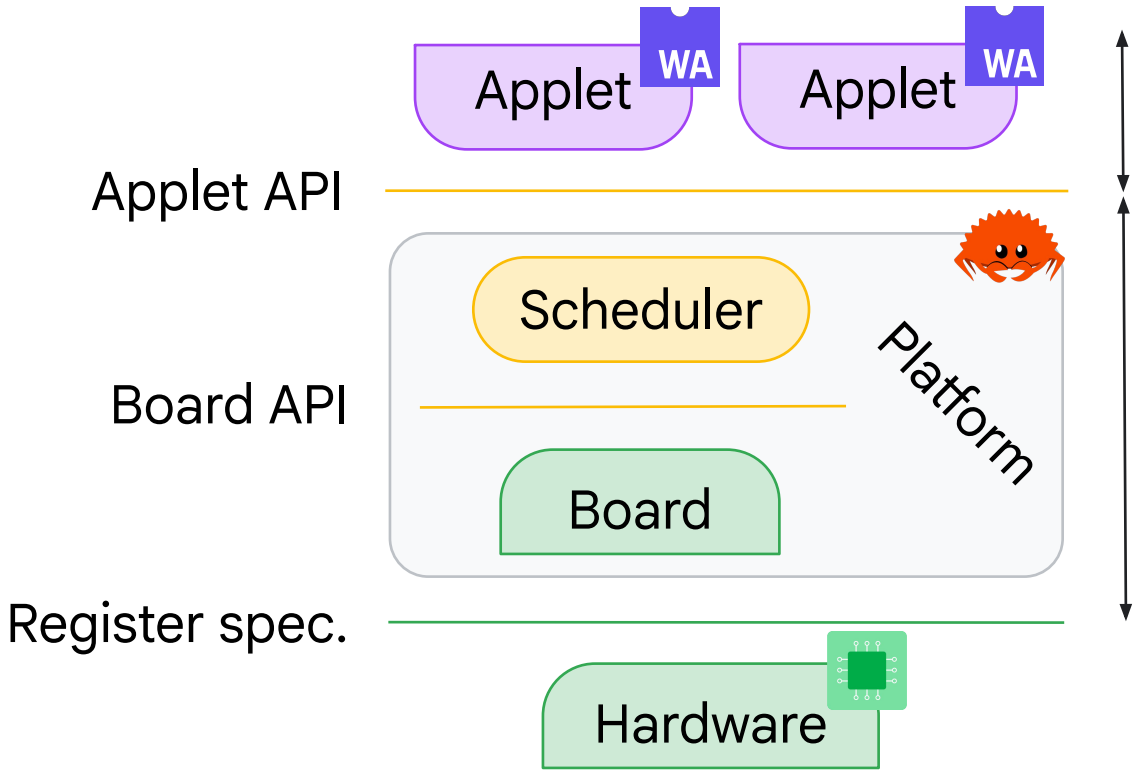
WebAssembly virtual machine

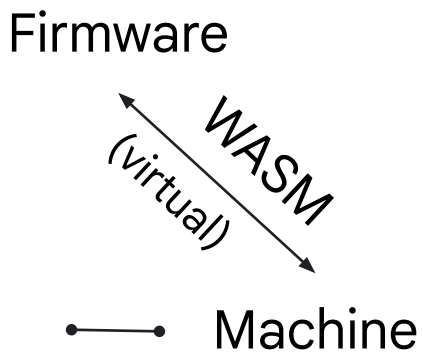


WebAssembly virtual machine

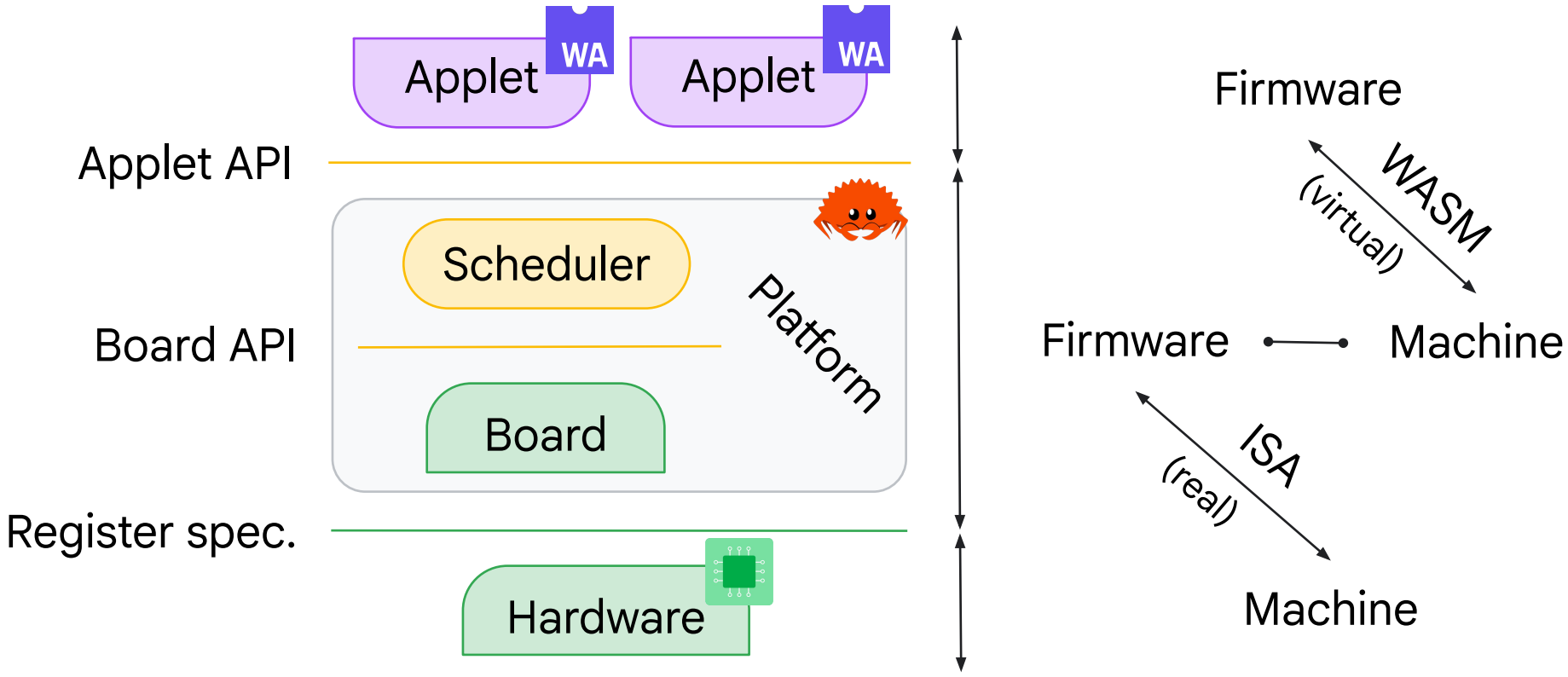


WebAssembly virtual machine





WebAssembly virtual machine



Applet and board APIs

Communication

GPIO, LED, buttons, UART, SPI, USB, BLE, ...

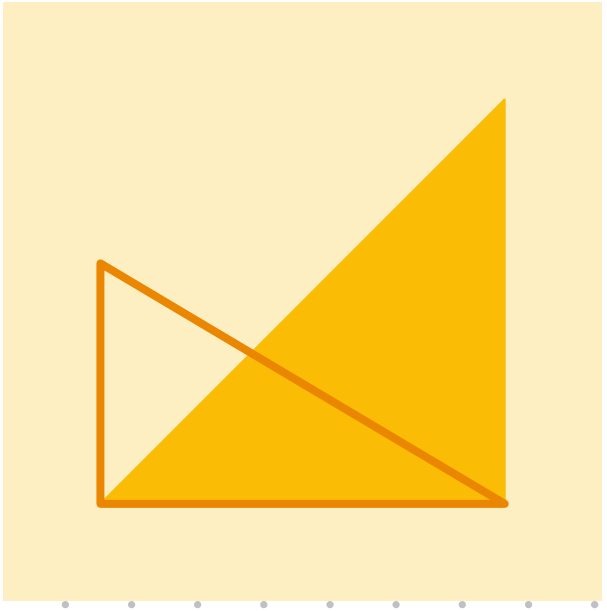
Cryptography

TRNG, DRBG, AES, SHA, HMAC, ECC, PQC, ...

Foundational toolkit

Debug output, perf. measurements, timers, storage, ...

Existing solutions



Existing solutions are either

Not secure

Existing solutions are not always secure by default

Existing solutions are either

Not secure

Existing solutions are not always secure by default

Wasefire is secure by design

Existing solutions are either

Not secure

Existing solutions are not always secure by default

Not accessible

Existing solutions target embedded developers

Wasefire is secure by design

Existing solutions are either

Not secure

Existing solutions are not always secure by default

Wasefire is secure by design

Not accessible

Existing solutions target embedded developers

Wasefire targets software engineers

Existing solutions are either

Not secure

Existing solutions are not always secure by default

Wasefire is secure by design

Not accessible

Existing solutions target embedded developers

Wasefire targets software engineers

Not portable

Existing solutions have at best source-level portability

Existing solutions are either

Not secure

Existing solutions are not always secure by default

Wasefire is secure by design

Not accessible

Existing solutions target embedded developers

Wasefire targets software engineers

Not portable

Existing solutions have at best source-level portability

Wasefire provides binary-level portability

But in comparison, Wasefire is

Not as performant

WebAssembly interpretation is 10x to 100x slower than native code

But in comparison, Wasefire is

Not as performant

WebAssembly interpretation is 10x to 100x slower than native code

Will improve with CHERI, Pulley, and/or regular hardware protection

But in comparison, Wasefire is

Not as performant

WebAssembly interpretation is 10x to 100x slower than native code

Not as complete

Couple supported boards, missing features, reduced API

Will improve with CHERI, Pulley, and/or regular hardware protection

But in comparison, Wasefire is

Not as performant

WebAssembly interpretation is 10x to 100x slower than native code

Will improve with CHERI, Pulley, and/or regular hardware protection

Not as complete

Couple supported boards, missing features, reduced API

Being open-source, development can scale through contributions

But in comparison, Wasefire is

Not as performant

WebAssembly interpretation is 10x to 100x slower than native code

Will improve with CHERI, Pulley, and/or regular hardware protection

Not as complete

Couple supported boards, missing features, reduced API

Being open-source, development can scale through contributions

Not even stable

Wasefire is still an experimental research project at this point

But in comparison, Wasefire is

Not as performant

WebAssembly interpretation is 10x to 100x slower than native code

Will improve with CHERI, Pulley, and/or regular hardware protection

Not as complete

Couple supported boards, missing features, reduced API

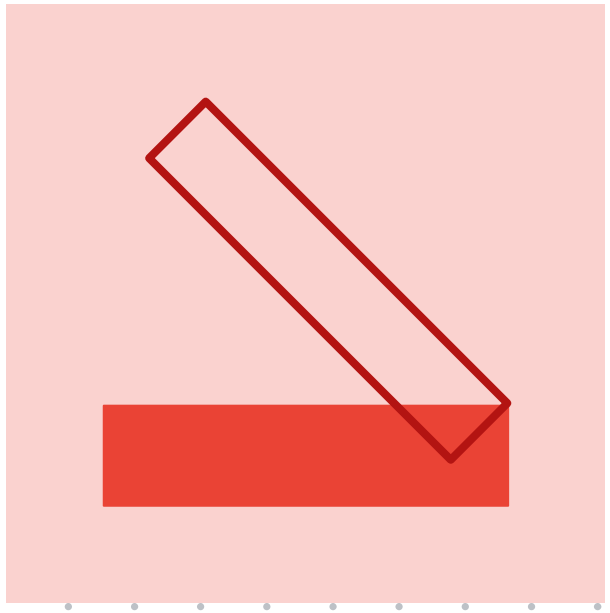
Being open-source, development can scale through contributions

Not even stable

Wasefire is still an experimental research project at this point

Stability will come as the user base grows

Current state and future work



Where we are

Boards	Nordic nRF52840, internal SE, OpenTitan Host with web UI
Applet Languages	Rust and AssemblyScript (low-level API only)
Scheduler	Currently limited to running 1 applet on 1 core Multi-applets and multi-core support to be added
Platform	Upgradability and applet management Hardening coming soon

Already being used!

Used internally

Two internal projects

2 research projects with universities

Improving hardware performance and security

Ongoing work for a commercial device

Result will be a hardware device powered by Wasefire

Wasefire for you

Google

Security and Privacy Research

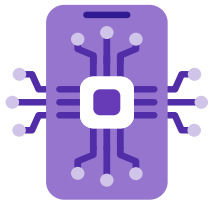
Optimal use-cases

No embedded background

Engineers can write firmware like software

Light interactive logic

Wasm applets cannot run heavy batch computations



Resource constrained device

Linux might be preferable otherwise

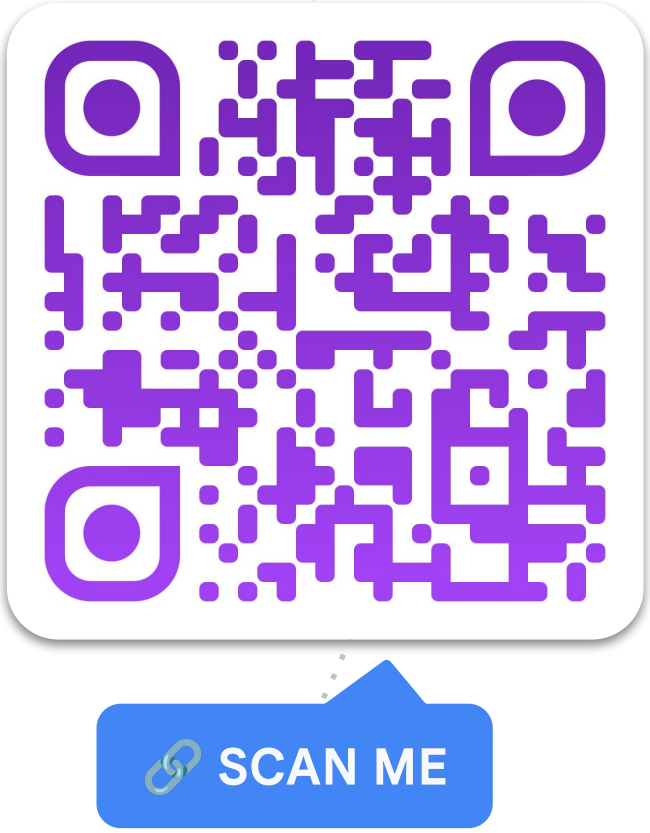
Assistance

Adding support for a new language

Adding support for a new board

Extending the applet API

Implementing security features



Documentation, tutorial and code available here:

https://google.github.io/wasefire/

Takeaways

Wasefire is still under active development but is already used

Takeaways

Wasefire is still under active development but is already used

Contributions and collaborations are welcome

Takeaways

Wasefire is still under active development but is already used

Contributions and collaborations are welcome

Looking to onboard an industrial use case in 2025

CYBER ON BOARD

Questions ?

CyberOnBoard 2025