Security for the cloud-edge continuum

There is no safety without security

Ischia, June 2025

Prof. P. Felber
University of Neuchâtel
pascal.felber@unine.ch

VEDLIoT | Very Efficient Deep Learning in IoT

vedliot.eu

INDUSTRIAL

The industrial use case employs DL-based solutions for motor condition monitoring and for arc detection, involving different challenges concerning the usage of DL models. In the former case it is necessary to comply with a ultra-low energy budget, and in the latter it is necessary to ensure a very low false-negative error rate.

AUTOMOTIVE

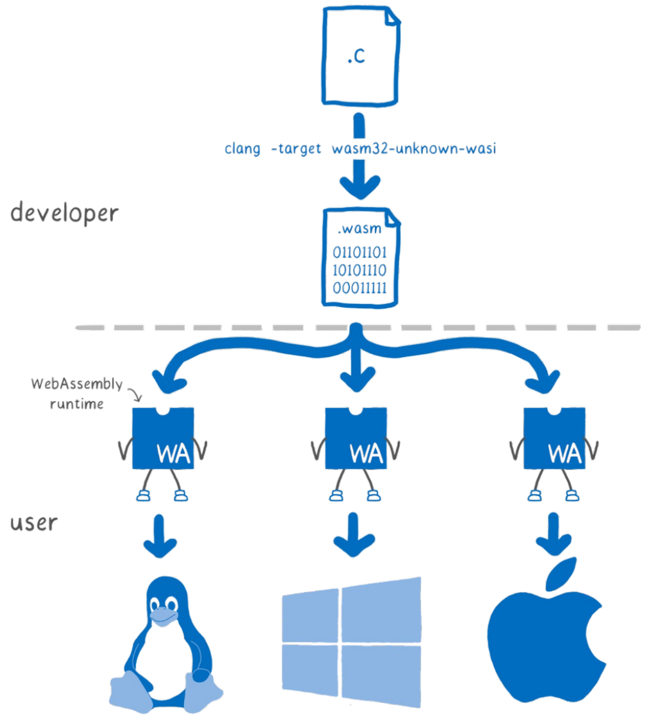
The automotive use-case focuses on increasing the processing efficiency DL tasks over the resources that are present in the traffic environment. This will be achieved through distribution of the processing tasks over resources such as the ego vehicle, cellular base station(s) in the close proximity, as well as the cloud.

HOME

In the home domain, VEDLIoT will consider a virtual mirror application in which it will be necessary to deploy the software and execute DNN models on different kinds of hardware, namely at the edge (t.RECS) and on a specialized embedded platform (µ.RECS).

Security for cloud-edge continuum [VEDLIoT]

- Wasm: standard for a bytecode format
  - Compilation target for mainstream programming languages
  - Universal runtime (not only for the web)
  - WebAssembly system interface (WASI) for system interactions
- Pros
  - **Lightweight** bytecode and specifications
  - Code execution is **sandboxed** (also protects the host)
  - Near-native **speed** with AOT and JIT compilation
  - **Same code** on cloud, edge, IoT devices: *cloud-edge continuum*



https://github.com/JamesMenetrey/unine-twine https://github.com/JamesMenetrey/unine-watz

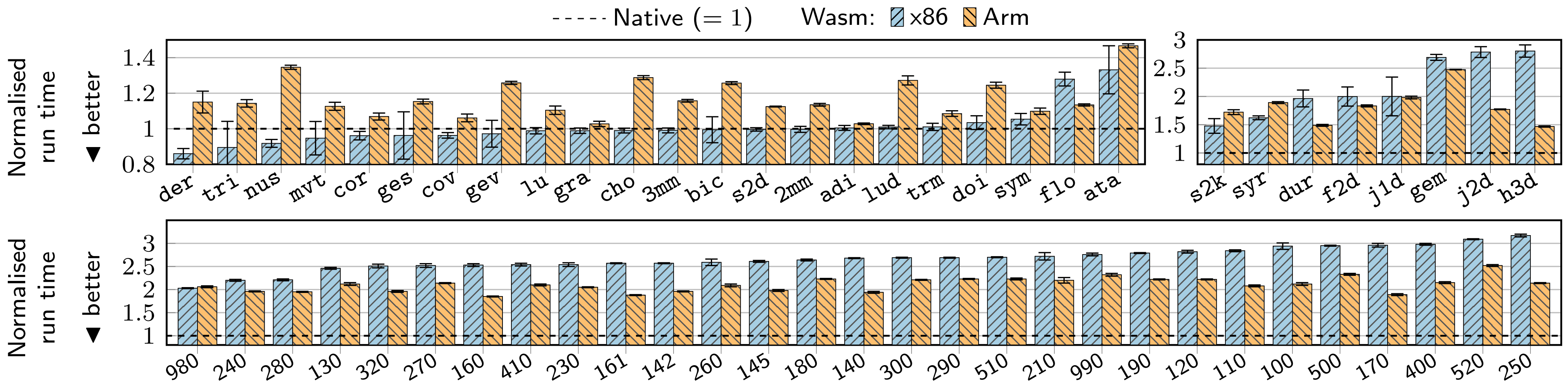
WebAssembly + TEEs

[VEDLIoT]

Twine for Intel SGX [ICDE'21] + WaTZ for Arm TrustZone [ICDCS'22]

- Execute Wasm code securely within TEE
- Leverage WASI to replace POSIX and deliver TEE features
- Benchmarks (Polybench/C and SQLite) show <3 slowdown

⇒ Confidential computing for the could-edge continuum



https://github.com/JamesMenetrey/unine-twine https://github.com/JamesMenetrey/unine-watz

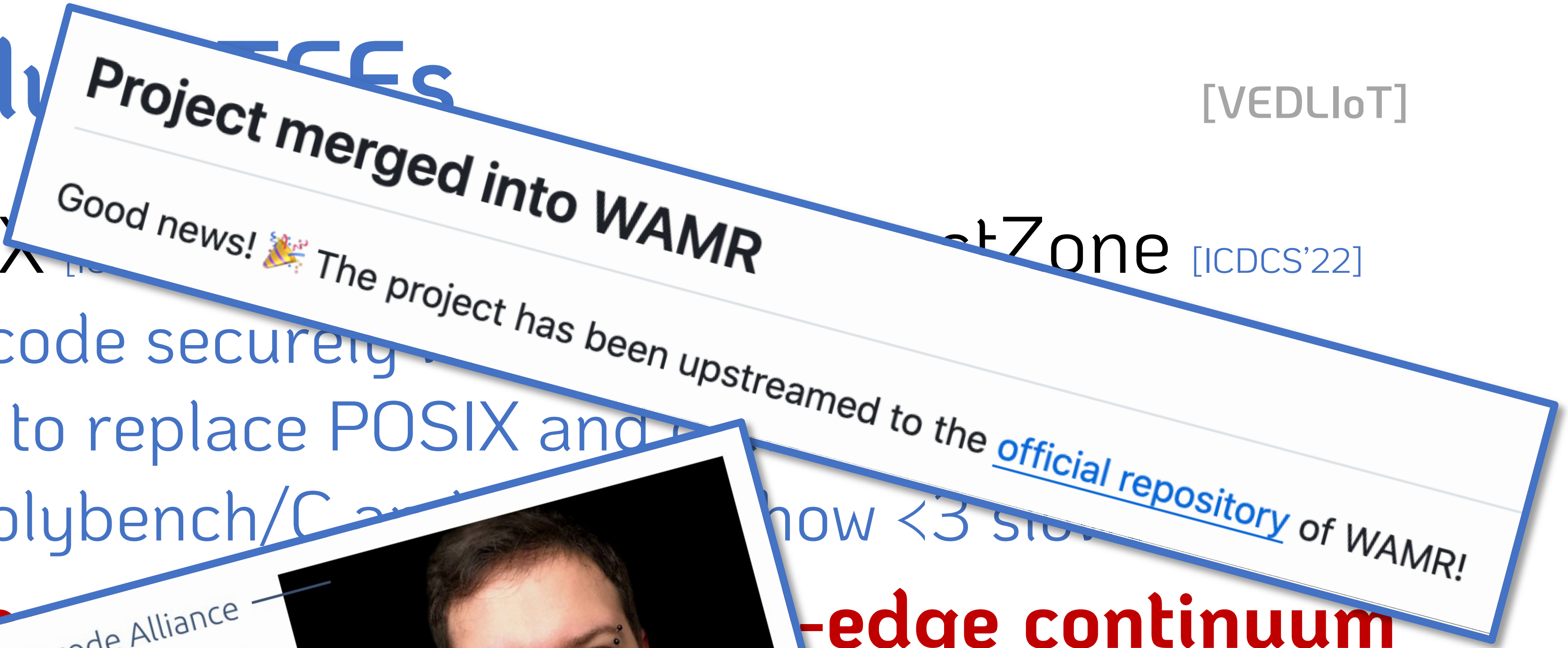
WebAssembl TEEs

[VEDLIoT]

Twine for Intel SGX [I tZone [ICDCS'22]

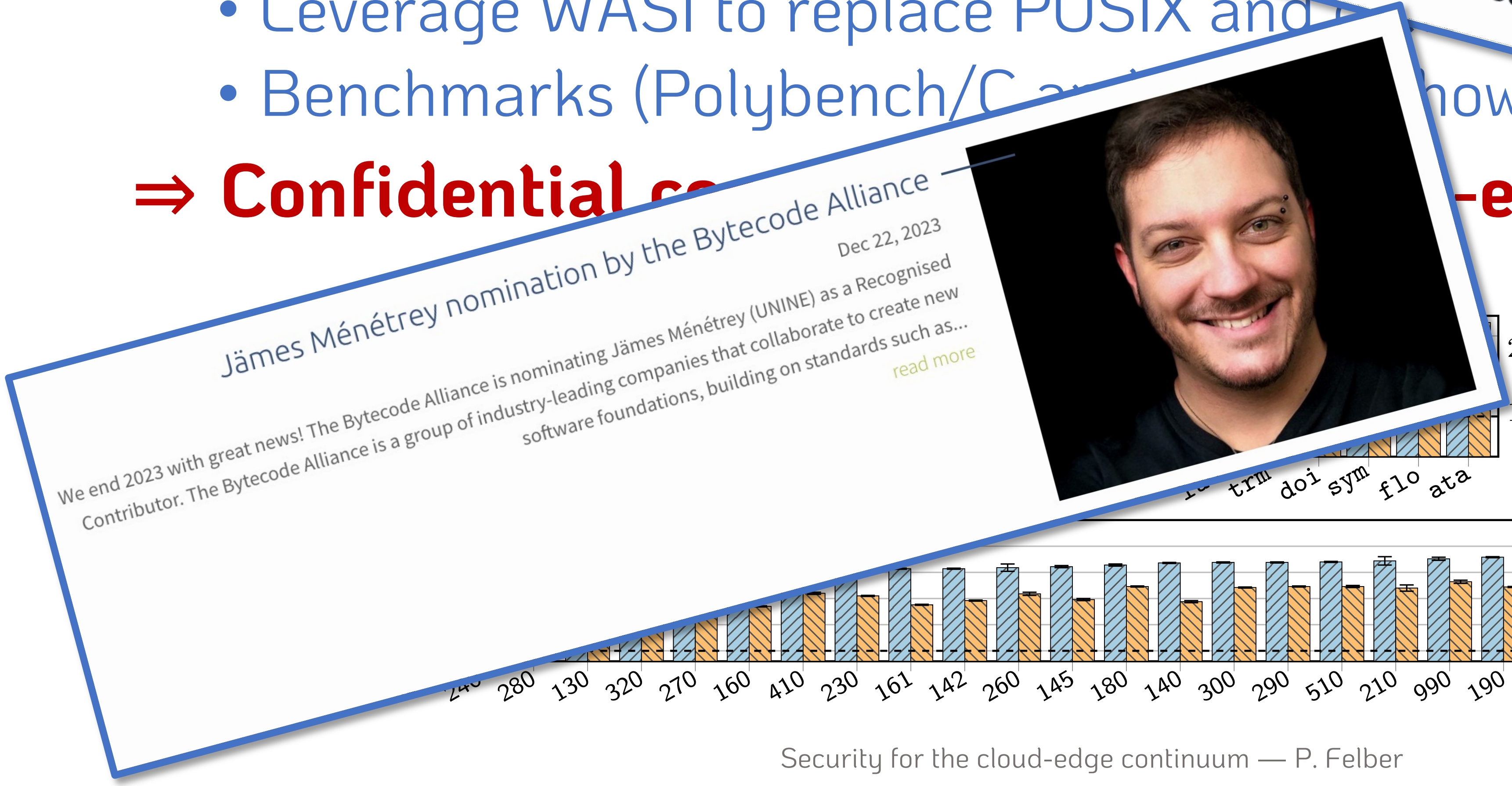
- Execute Wasm code securely
- Leverage WASI to replace POSIX and
- Benchmarks (Polybench/C a how <3 slo

⇒ Confidential c -edge continuum



Project merged into WAMR

Good news! 🎉 The project has been upstreamed to the official repository of WAMR!

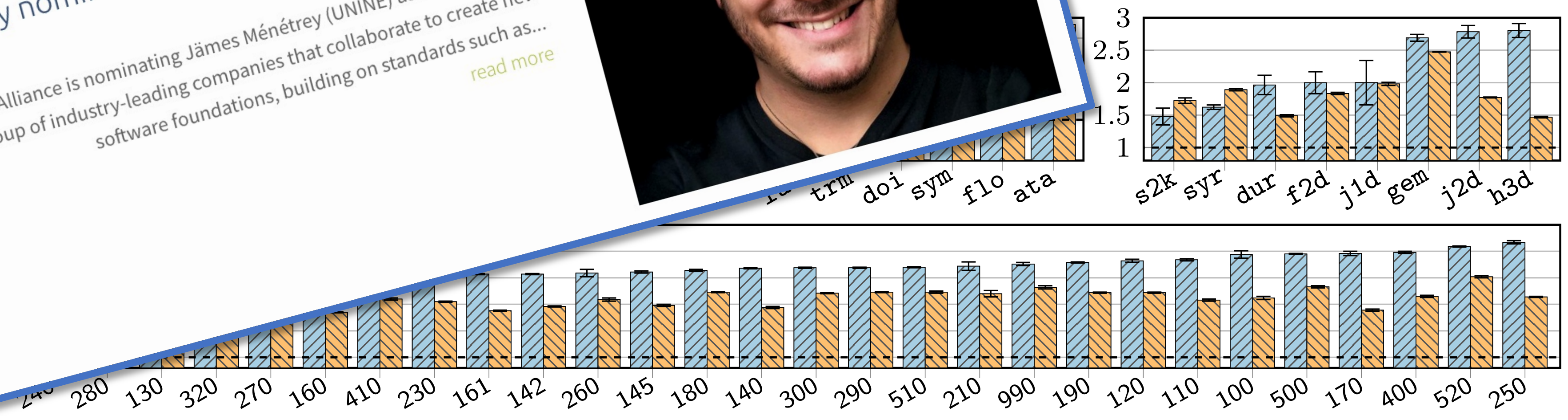


Jämes Ménétrey nomination by the Bytecode Alliance

Dec 22, 2023

We end 2023 with great news! The Bytecode Alliance is nominating Jämes Ménétrey (UNINE) as a Recognised Contributor. The Bytecode Alliance is a group of industry-leading companies that collaborate to create new software foundations, building on standards such as...

read more



Moving forward

TEEs are no silver bullet…

- Require craft from programmers, knowledge of system issues
- Might lack fundamental properties (e.g., attestation)
- Performance can be poor (e.g., memory limitations)
- Continuous stream of (side-channel) attacks

…and decentralisation is not always the best strategy…

- Sometimes centralisation might be safer [1st day presentations]

…***but*** there is no safety without security

- TEEs add (some) security, WASM adds portability/simplicity

⇒ *We leverage **Twine/WaTZ** to secure **supply-chain** management systems and **smart contracts/DApps*** [SCEAL]

Extra slides

Trusted execution environments (TEEs)

- **TEEs** isolate applications from the rest of the system
  - Segregated area of memory and CPU protected by HW against powerful attacks
  - Its content is shielded from other applications, compromised OS and system libraries, attackers with physical access to the machine, …
- Uses “attestation” to verify SW and HW before execution
- Guarantees **data confidentiality** and **code integrity**
  - Prevents unauthorised parties outside TEE from reading data
  - Prevents unauthorised parties from replacing or modifying code in TEE

Trusted execution environments (TEEs)

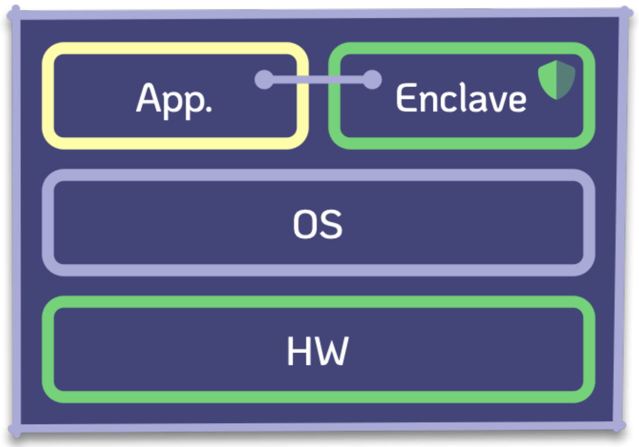
- Various TEE architectures exist and depend on the CPU
- They differ by their threat model and capabilities
  - **Intel SGX:** enclaves
  - **Arm TrustZone:** separate systems (two “worlds”)
  - **AMD SEV:** virtualised systems (VMs)
  - **Intel TDX:** trusted domains (VMs)
  - **Arm CCA:** realms (system-wide hardware isolation)
  - **RISC-V:** several proposed extensions

  …

Intel SGX

Software guard extensions

- Hardware extension in recent Intel CPUs since Skylake (2015)
- Protects confidentiality and integrity of code and data in untrusted environments
  - The platform is considered malicious by default
  - Only the CPU chip and the isolated region are trusted
  - Code is attested (via Intel attestation service)
- Code runs in an “enclave”: a piece of trusted software

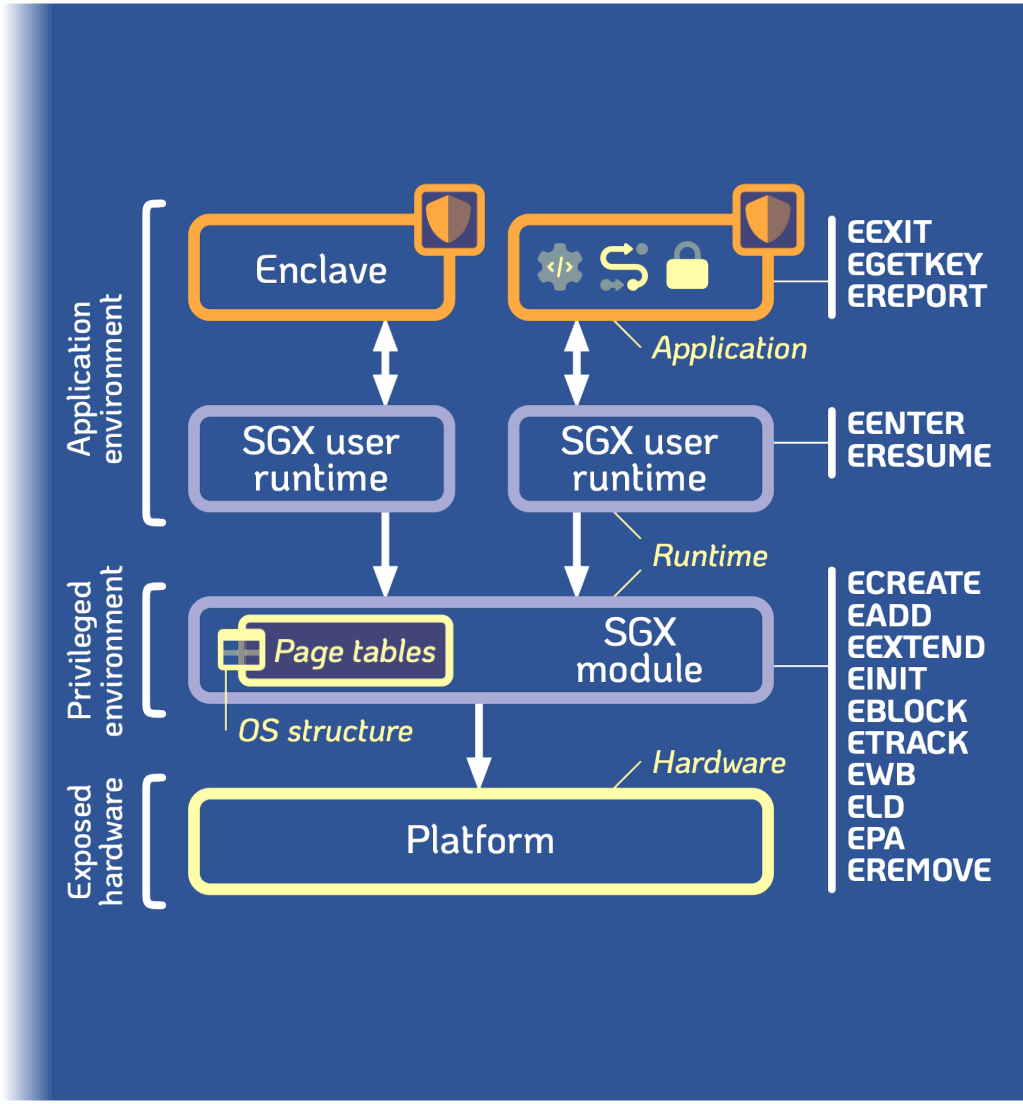


SGX architecture and API

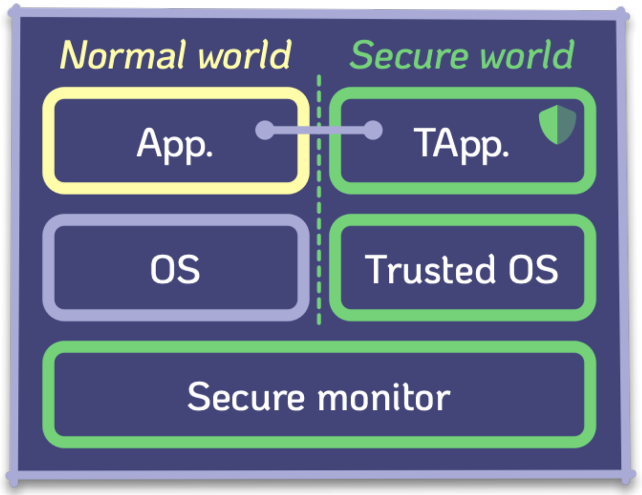
- Secure code runs "native speed"…

  …but API is quite complex
  - Need to heavily modify legacy code

  …small enclave page cache (EPC)
  - SGXv1: 128 MB (~96 MB w/out paging)
  - SGXv2: up to 1 TB
- Performance of memory accesses
  - Native speed in L1/L2/L3 cache
  - Reasonable within the EPC
  - Huge when paging to main memory



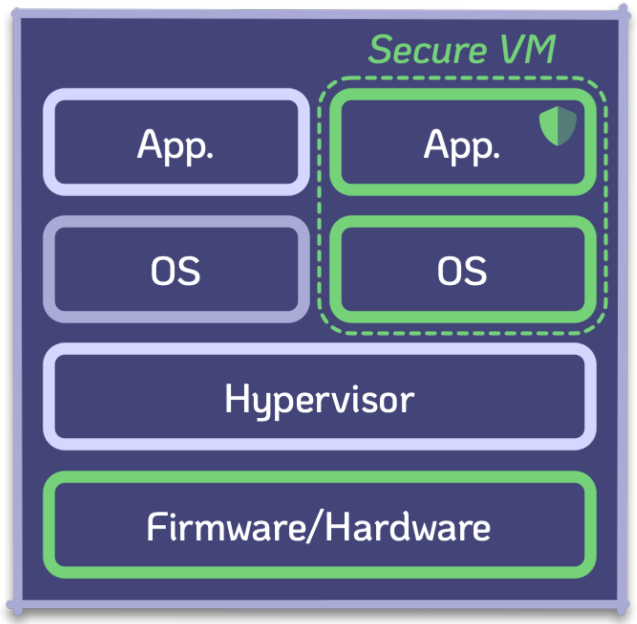
Arm TrustZone (TZ)



- TZ is widely spread on small and IoT devices with a Cortex-A/M processor
- Separates devices in two worlds
  - The **normal** world
  - The **secure** world
- One trusted application (TA) at a time
- Provides memory confidentiality but not integrity
- No built-in attestation service
- Limited memory per TA (~4–32 MB in practice)

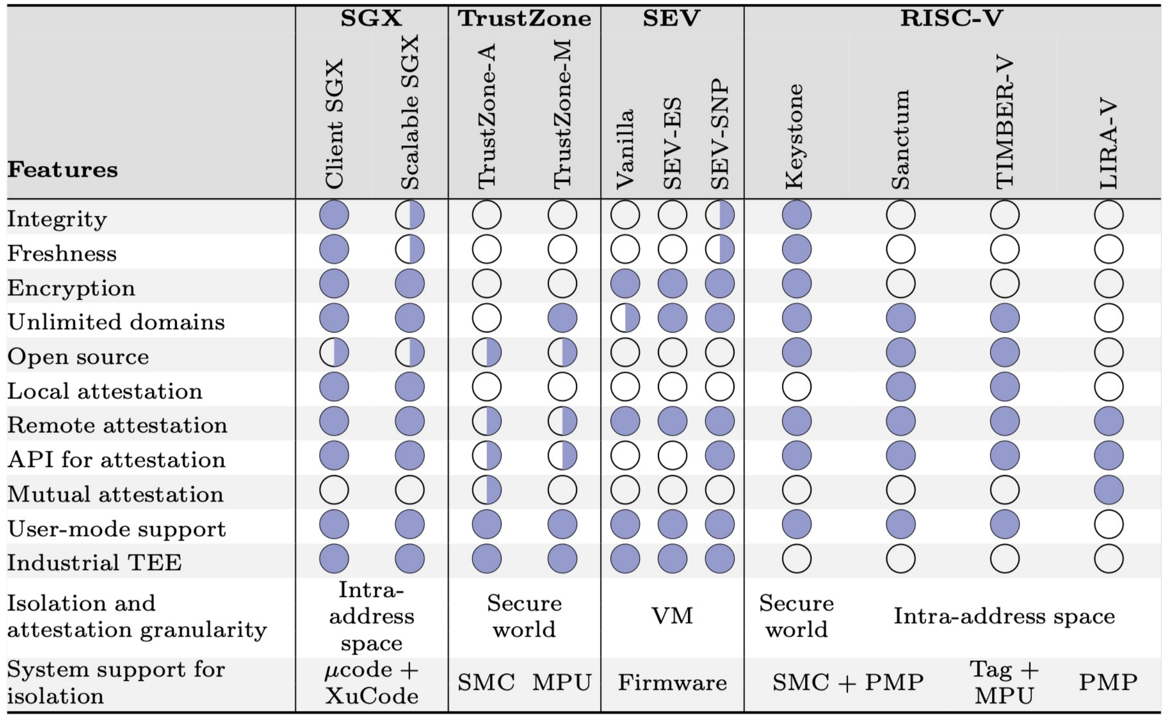
AMD SEV

- SEV-SNP is supported on computers and servers with EPYC 7003+ series processors
- Each trusted environment is a secure virtual machine
- SEV-SNP provides both memory confidentiality and integrity
- Support for remote attestation
- Unlimited amount of addressable memory



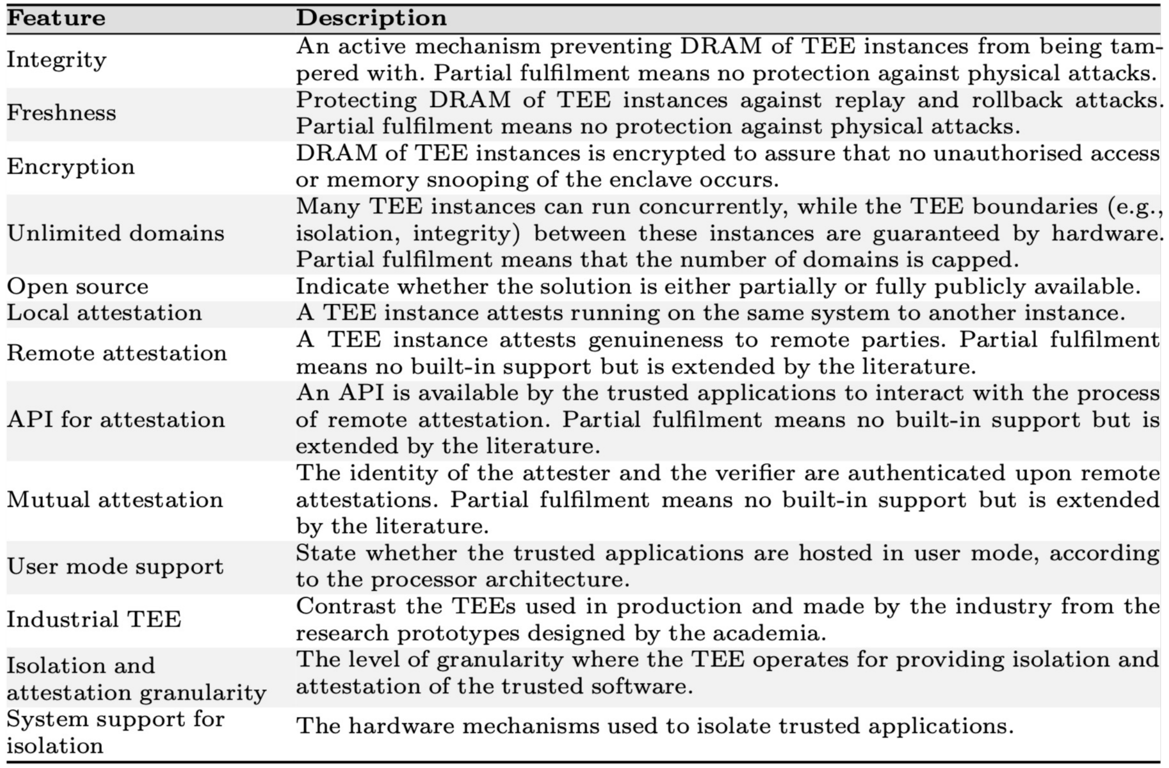
Comparison of TEEs

[https://arxiv.org/pdf/2206.03780]



Features	SGX		TrustZone		SEV			RISC-V			
	Client SGX	Scalable SGX	TrustZone-A	TrustZone-M	Vanilla	SEV-ES	SEV-SNP	Keystone	Sanctum	TIMBER-V	LIRA-V
Integrity	●	◐	○	○	○	○	◐	●	○	○	○
Freshness	●	◐	○	○	○	○	◐	●	○	○	○
Encryption	●	●	○	○	●	●	●	●	○	○	○
Unlimited domains	●	●	○	●	◐	●	●	●	●	●	○
Open source	◐	◐	◐	◐	○	○	○	●	●	●	○
Local attestation	●	●	○	○	○	○	○	○	●	●	○
Remote attestation	●	●	◐	◐	●	●	●	●	●	●	●
API for attestation	●	●	◐	◐	○	○	●	●	●	●	●
Mutual attestation	○	○	◐	○	○	○	○	○	○	○	●
User-mode support	●	●	●	●	●	●	●	●	●	●	○
Industrial TEE	●	●	●	●	●	●	●	○	○	○	○
Isolation and attestation granularity	Intra-address space		Secure world		VM			Secure world	Intra-address space		
System support for isolation	μcode + XuCode		SMC	MPU	Firmware			SMC + PMP		Tag + MPU	PMP

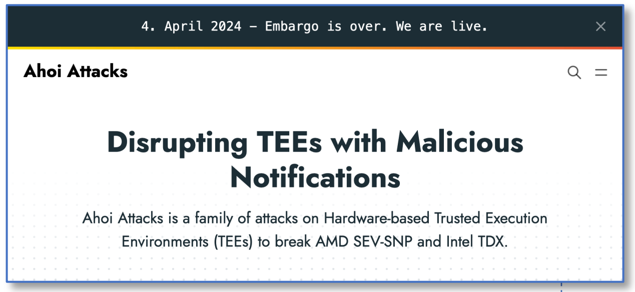
Table 1: Comparison of the state-of-the-art TEEs.



Feature	Description
Integrity	An active mechanism preventing DRAM of TEE instances from being tampered with. Partial fulfilment means no protection against physical attacks.
Freshness	Protecting DRAM of TEE instances against replay and rollback attacks. Partial fulfilment means no protection against physical attacks.
Encryption	DRAM of TEE instances is encrypted to assure that no unauthorised access or memory snooping of the enclave occurs.
Unlimited domains	Many TEE instances can run concurrently, while the TEE boundaries (e.g., isolation, integrity) between these instances are guaranteed by hardware. Partial fulfilment means that the number of domains is capped.
Open source	Indicate whether the solution is either partially or fully publicly available.
Local attestation	A TEE instance attests running on the same system to another instance.
Remote attestation	A TEE instance attests genuineness to remote parties. Partial fulfilment means no built-in support but is extended by the literature.
API for attestation	An API is available by the trusted applications to interact with the process of remote attestation. Partial fulfilment means no built-in support but is extended by the literature.
Mutual attestation	The identity of the attester and the verifier are authenticated upon remote attestations. Partial fulfilment means no built-in support but is extended by the literature.
User mode support	State whether the trusted applications are hosted in user mode, according to the processor architecture.
Industrial TEE	Contrast the TEEs used in production and made by the industry from the research prototypes designed by the academia.
Isolation and attestation granularity	The level of granularity where the TEE operates for providing isolation and attestation of the trusted software.
System support for isolation	The hardware mechanisms used to isolate trusted applications.

Table 2: Features of the state-of-the-art TEEs.

TEEs are no silver bullet



4. April 2024 – Embargo is over. We are live.

Ahoi Attacks

Disrupting TEEs with Malicious Notifications

Ahoi Attacks is a family of attacks on Hardware-based Trusted Execution Environments (TEEs) to break AMD SEV-SNP and Intel TDX.

- Require some craft from programmers
  - SDK is only available for limited programming languages
  - Constrained development environments
- Might lack fundamental properties
  - E.g., attestation or integrity are not always supported
- Performance can be poor (e.g., memory limitations)
- Requires good knowledge of system issues
  - No POSIX API (hard to write or migrate existing applications)
- Continuous stream of (side-channel) attacks