Enarx

Protection for data in use

Mike Bursell
Office of the CTO

https://enarx.io

Nathaniel McCallum
Sr. Principal Engineer

Red Hat

Trusted Execution Environments

Red Hat

Trusted Execution Environments

TEE is a protected area within the host, for execution of sensitive workloads

Trusted Execution Environments

TEE is a protected area within the host, for execution of sensitive workloads

TEE provides:

- Memory Confidentiality
- Integrity Protection
- General compute
- HWRNG

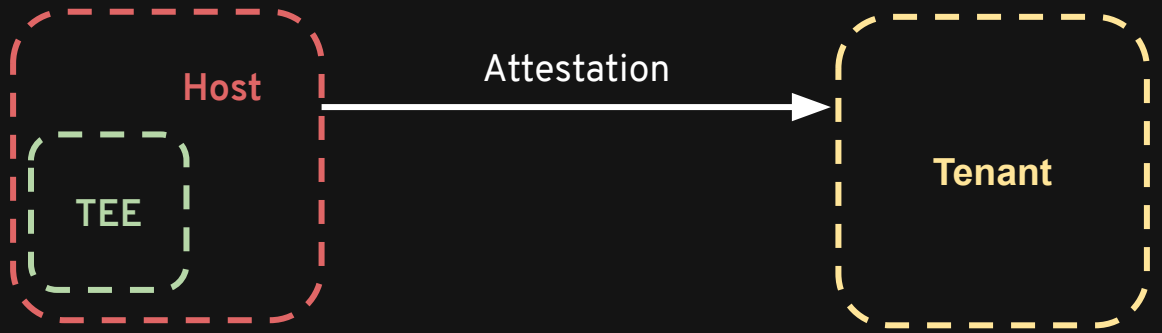
Trusted Execution Environments

Q. “But how do I know that it’s a valid TEE?”

TEE provides:

- Memory Confidentiality
- Integrity Protection
- General compute
- HWRNG

Trusted Execution Summary

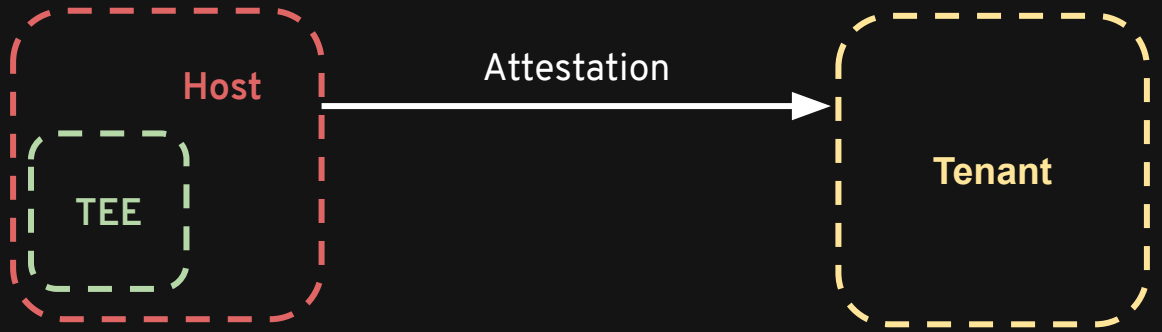


Q. “But how do I know that it’s a valid TEE?”
A. Attestation

TEE provides:

- Memory Confidentiality
- Integrity Protection
- General compute
- HWRNG

Trusted Execution Summary



Attestation includes:

- Diffie-Hellman Public Key
- Hardware Root of Trust
- TEE Measurement

TEE provides:

- Memory Confidentiality
- Integrity Protection
- General compute
- HWRNG

Trusted Execution Summary

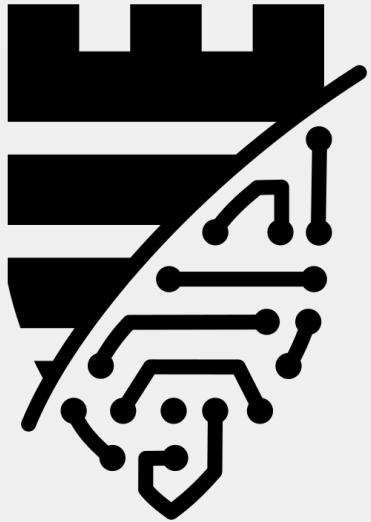
Attestation includes:

- Diffie-Hellman Public Key
- Hardware Root of Trust
- TEE Measurement

TEE provides:

- Memory Confidentiality
- Integrity Protection
- General compute
- HWRNG

Introducing Enarx



Red Hat

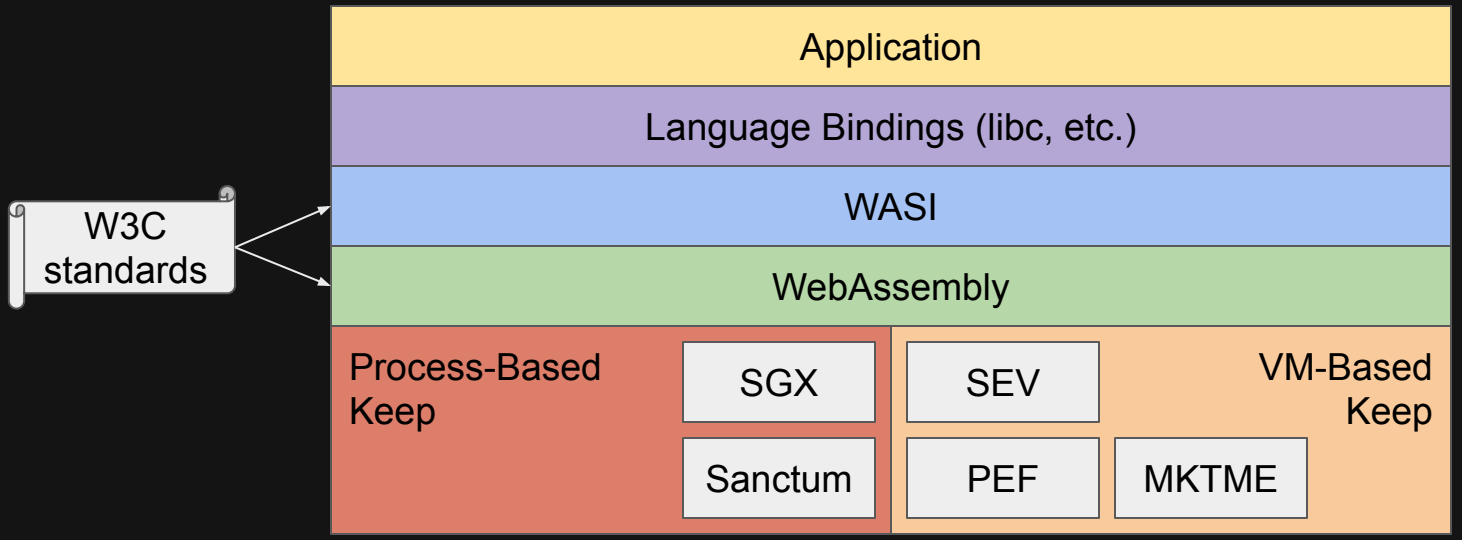
Enarx Principles

1. We don’t trust the host owner
2. We don’t trust the host software
3. We don’t trust the host users
4. We don’t trust the host hardware
    a. … but we’ll make an exception for CPU + firmware

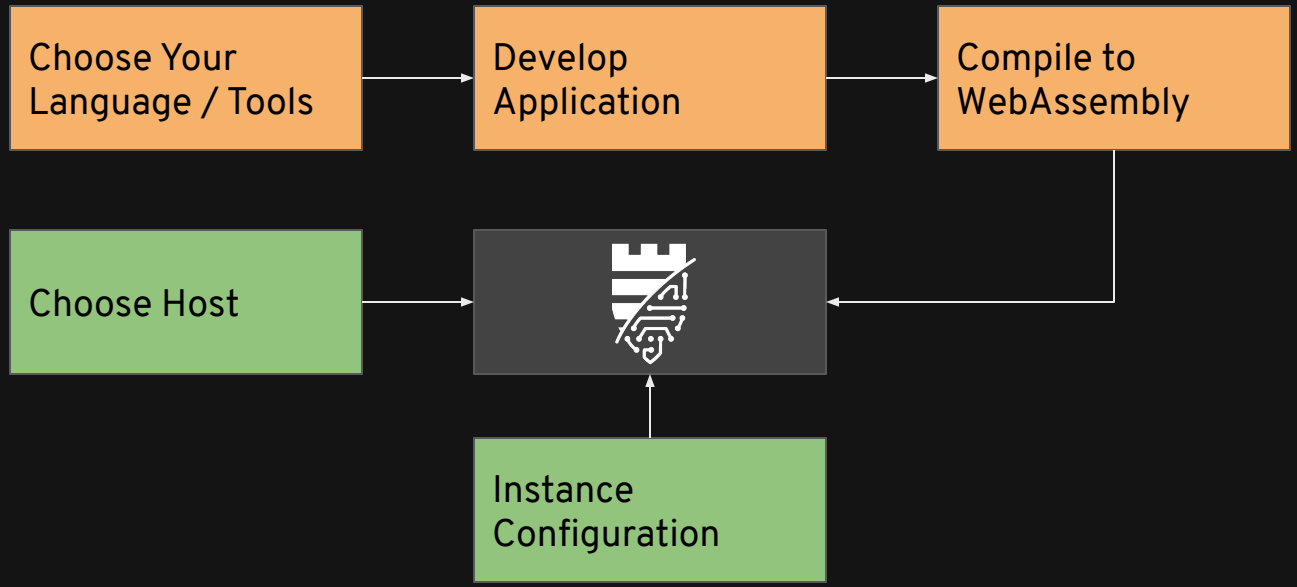
Enarx Design Principles

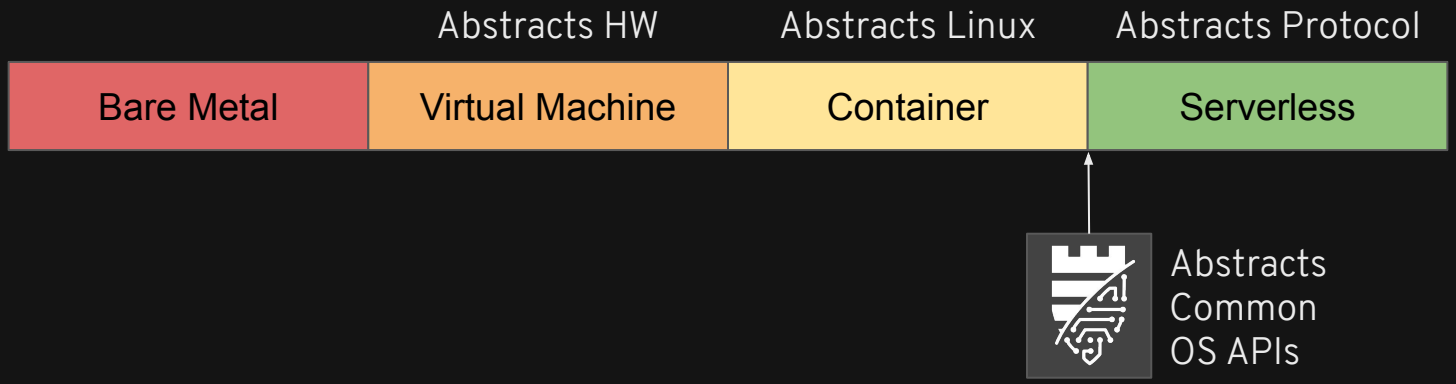
1. Minimal Trusted Computing Base
2. Minimum trust relationships
3. Deployment-time portability
4. Network stack outside TCB
5. Security at rest, in transit and in use
6. Auditability
7. Open source
8. Open standards
9. Memory safety
10. No backdoors

Enarx Architecture



Enarx is a ~~Development~~ Deployment Framework

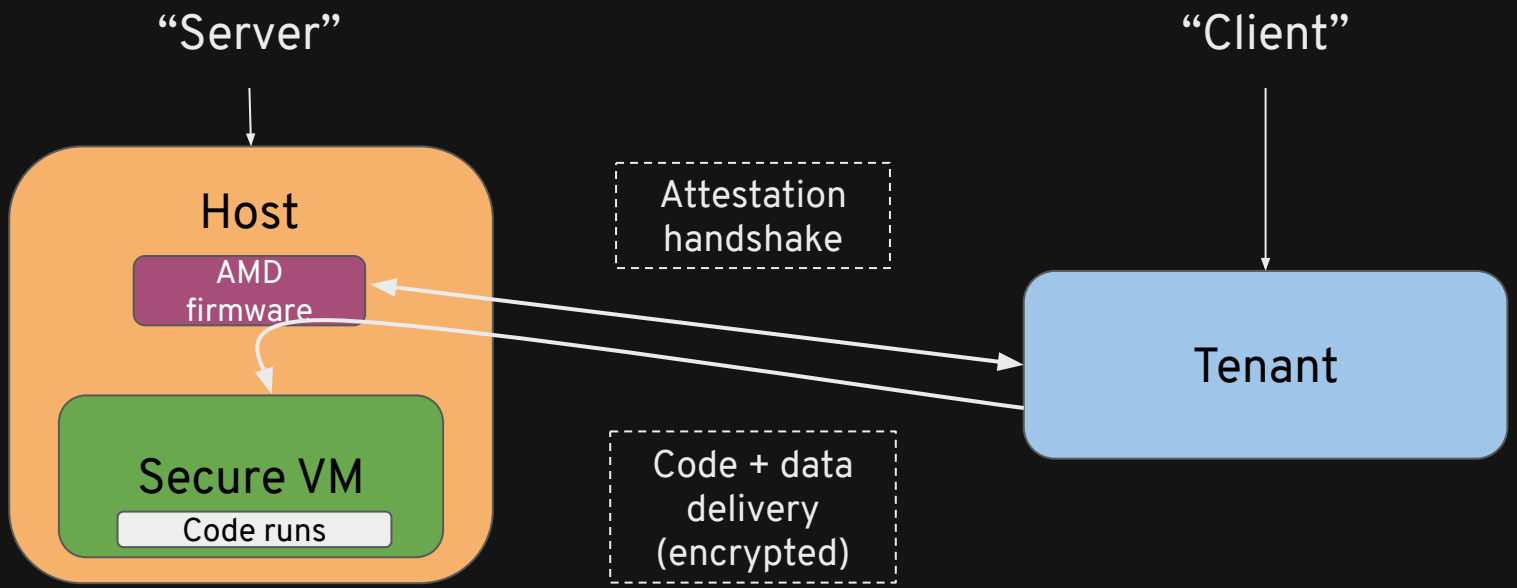




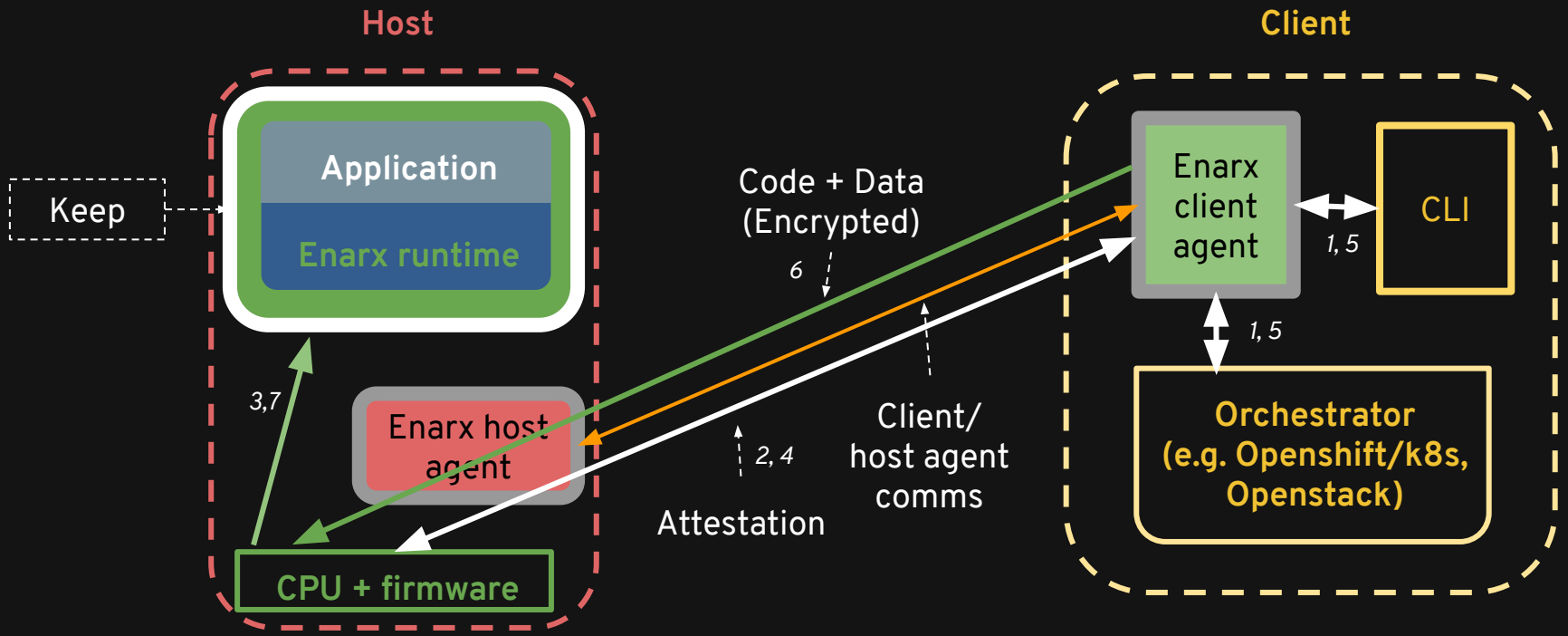
Just enough legacy support to enable trivial application portability.
Homogeneity to enable radical deployment-time portability.
No interfaces which accidentally leak data to the host.
Bridges process-based and VM-based TEE models.
No operating system to manage.

Process flow

Overview (AMD example)

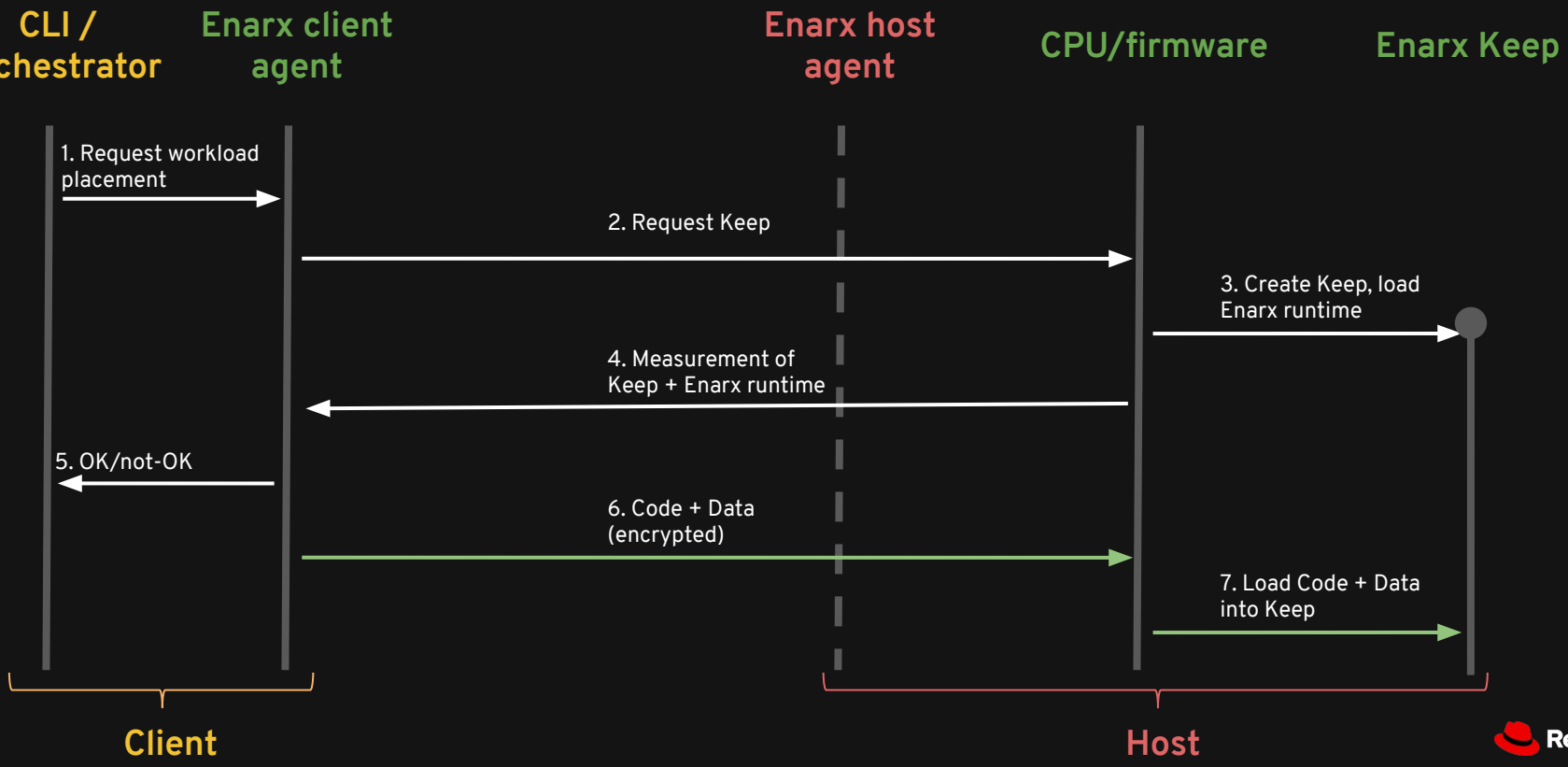


Enarx architectural components



Red Hat

Enarx attestation process diagram



Red Hat

Enarx Status

Red Hat

Current Status

1. SEV: Fully attested demo w/ custom assembly.
    a. Ketuvim: KVM library with SEV support
2. SGX: Fully attested demo w/ data delivery.
3. PEF: Ongoing discussions with POWER team.
4. WASM/WASI: Demo with some basic WASI functions.

We Need Your Help!

Website: https://enarx.io

Code: https://github.com/enarx

Master plan: https://github.com/enarx/enarx/issues/1

License: Apache 2.0

Language: Rust

Questions?

https://enarx.io

Red Hat