Open Enclave SDK

Dave Thaler

Confidential Computing Consortium

- CCC is part of the Linux Foundation
- Website: https://confidentialcomputing.io/
- Member orgs (some already have IETF TEEP/RATS/SUIT doc authors)
  - **Alibaba**, **ARM**, Facebook, Google, **Huawei**, **Intel**, **Microsoft**, Oracle, Red Hat, …
- CCC does implementation, marketing, etc., and coordinates with IETF, TCG, GP, etc. for protocols
- Open source projects accepted:
  - Enarx (submitted by Red Hat)
  - Open Enclave SDK (submitted by Microsoft)
  - SGX SDK for Linux (submitted by Intel)
  - …

CONFIDENTIAL COMPUTING CONSORTIUM

Trusted Execution Environments

- A TEE provides hardware-enforcement that
  - The device has unique security identity
  - Any code inside the TEE is authorized code
    - Reduced risk for application compromise
  - Any data inside the TEE cannot be read by code outside the TEE
    - Safe area of the device to protect assets (great for key management, ML models, etc.)
  - Compromising REE and normal apps don't affect TEE and code (called Trusted Applications) running inside TEE

- TEE Examples:
  - Intel SGX, ARM TrustZone, Secure Elements, RISC-V MultiZone, etc.

Two styles of Trusted App platforms exist

- LibOS/Microkernel style: Enarx, Graphene, etc.
  - Runs entire application inside a TEE, with some sort of OS under it in the TEE
  - Much higher Trusted Computing Base (higher security risk)
  - Lower dev effort

- Enclave style: Open Enclave SDK, Intel SGX SDK, etc.
  - Run only security critical functionality inside the TEE
  - Minimizes Trusted Computing Base (lower security risk)
  - Higher dev effort

TrustBox: CES 2019 winner for cybersecurity

- "CES Innovation Awards winners have been announced ahead of CES 2019, with the cybersecurity and personal privacy award going to Scalys' TrustBox, a router and IoT gateway designed to secure IoT devices on home networks."
- "Scalys worked with Microsoft and semiconductor manufacturer NXP to build TrustBox around the NXP Layerscape LS1012A networking processor, an SoC with hardware security features including secure boot, secure software provisioning, and secure storage. TrustBox also features Microsoft's open source Open Enclave SDK, which can make it flexible for businesses that need to develop trusted execution environments."

Trusted App development today is hard!

Far fewer Trusted Apps than normal apps today

Among other problems:

- Trusted App platforms today provide different APIs than are available for normal apps
- Emulator environments today aren't integrated with the development environment

SGX-specific notes

Each trusted app runs in its own SGX container called an “enclave”.

“Trusted Application” is a signed shared library loaded in a hardware-protected part of the same process

- **Absolute time:** SGX has no time-of-day clock, so you have to ask something outside SGX if you need the current time, or implement a secure time protocol inside SGX.
- **Relative time:** SGX can track the passage of time, but the granularity is in seconds, not more fine grained.
- **Timers:** SGX has no timer API, so an app must implement its own timer queue outside SGX and call into the enclave when a timer expires.
- **Synchronization:** SGX has spinlocks, event objects, and mutexes, but not semaphores or timed waits. That is, waiting for event objects requires waiting for either 0 or INFINITE time.
- **Threads:** SGX cannot create or delete threads. Threads must be created outside SGX and call into an enclave to do work.
- **I/O:** stdout/stdin APIs, and networking APIs, are not available inside SGX, since I/O is not trusted.
- **Files:** SGX file APIs allow accessing sealed storage files, but there's no way to enumerate such files from within SGX. File metadata (size, modification time, etc.) is all visible outside SGX.

OP-TEE (TrustZone) notes

OP-TEE is a open-source platform that runs inside TrustZone and hosts Trusted Apps, each in its own container.

“Trusted Application” is an executable binary running in the TEE, so separate process and separate “OS”

- **Absolute time:** OP-TEE has no secure time-of-day clock, so you have to implement a secure time protocol inside the TEE, or rely on the untrusted time-of-day API.
- **Relative time:** OP-TEE can track the passage of time, with millisecond granularity.
- **Timers:** OP-TEE only has a blocking wait API. For asynchronous timers, an app must implement its own timer queue outside the TEE and call into the Trusted App when a timer expires.
- **Synchronization:** OP-TEE has (blocking) timed waits, and exclusive access locks on persistent objects, but no real synchronization primitives.
- **Threads:** TEE code cannot create or delete threads. Threads must be created outside the TEE and call into the TEE to do work.
- **I/O:** stdout/stdin APIs, and networking APIs, are not available inside OP-TEE.
- **Files:** TEE code can call APIs to access sealed storage files, but there's no API to enumerate such files. File metadata (size, modification time, etc.) is *not* visible outside the TEE.

Open Enclave SDK Goals

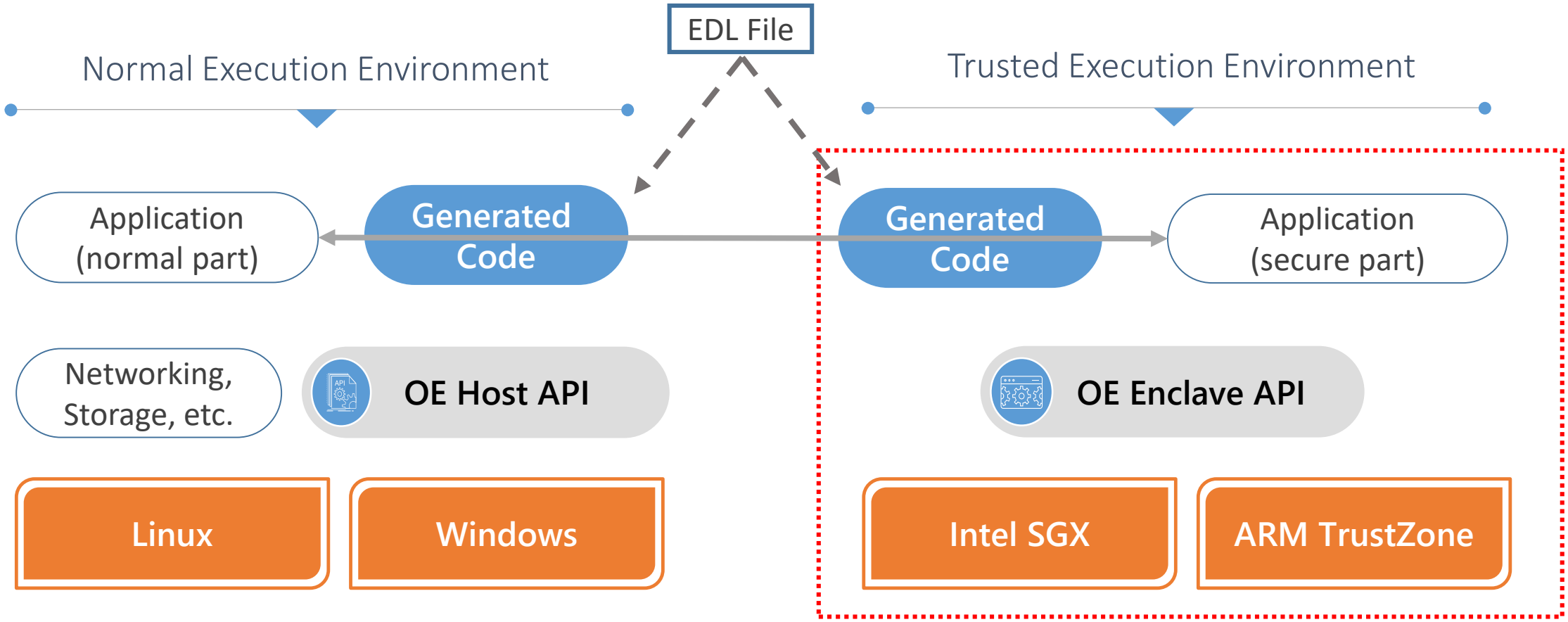
1. Easy migration from normal world app code to Trusted App code
2. Make it easy to write & debug new Trusted App code
3. Allow common app code for SGX, TrustZone, etc.
4. Be fully open source
5. Allow common app code independent of topological location (cloud vs. edge/IoT)
6. Allow attested communication between TZ/SGX/etc apps
7. Support both Linux and Windows hosts (and others)

Supported SDK functionality

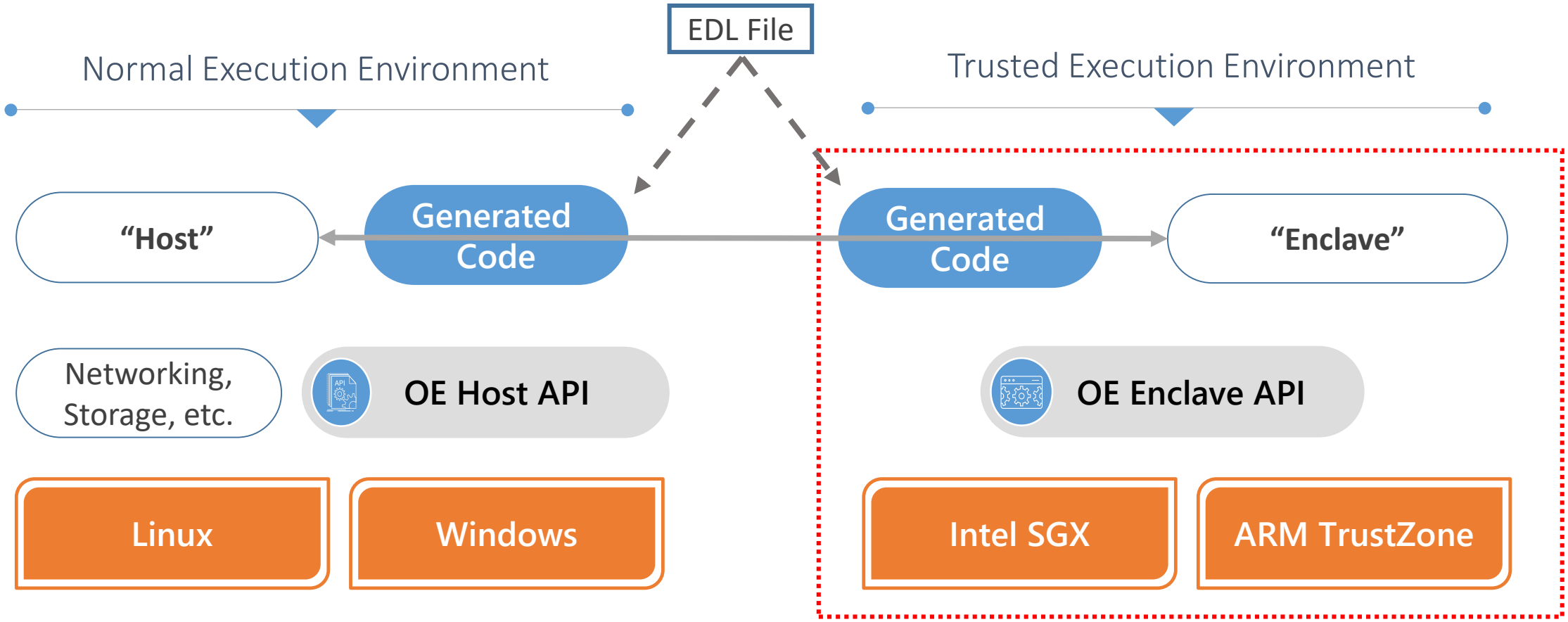
- Enclave creation and management
- Communication between app and enclave
- Sealing
- Cryptographic libraries
- Enclave measurement and identity
- Attestation

- POSIX APIs for enclaves:
  - Threads
  - Memory management
  - Files
  - Sockets
  - …
- Emulator support:
  - Simulation mode at runtime (ELF)
  - QEMU VM (TrustZone)

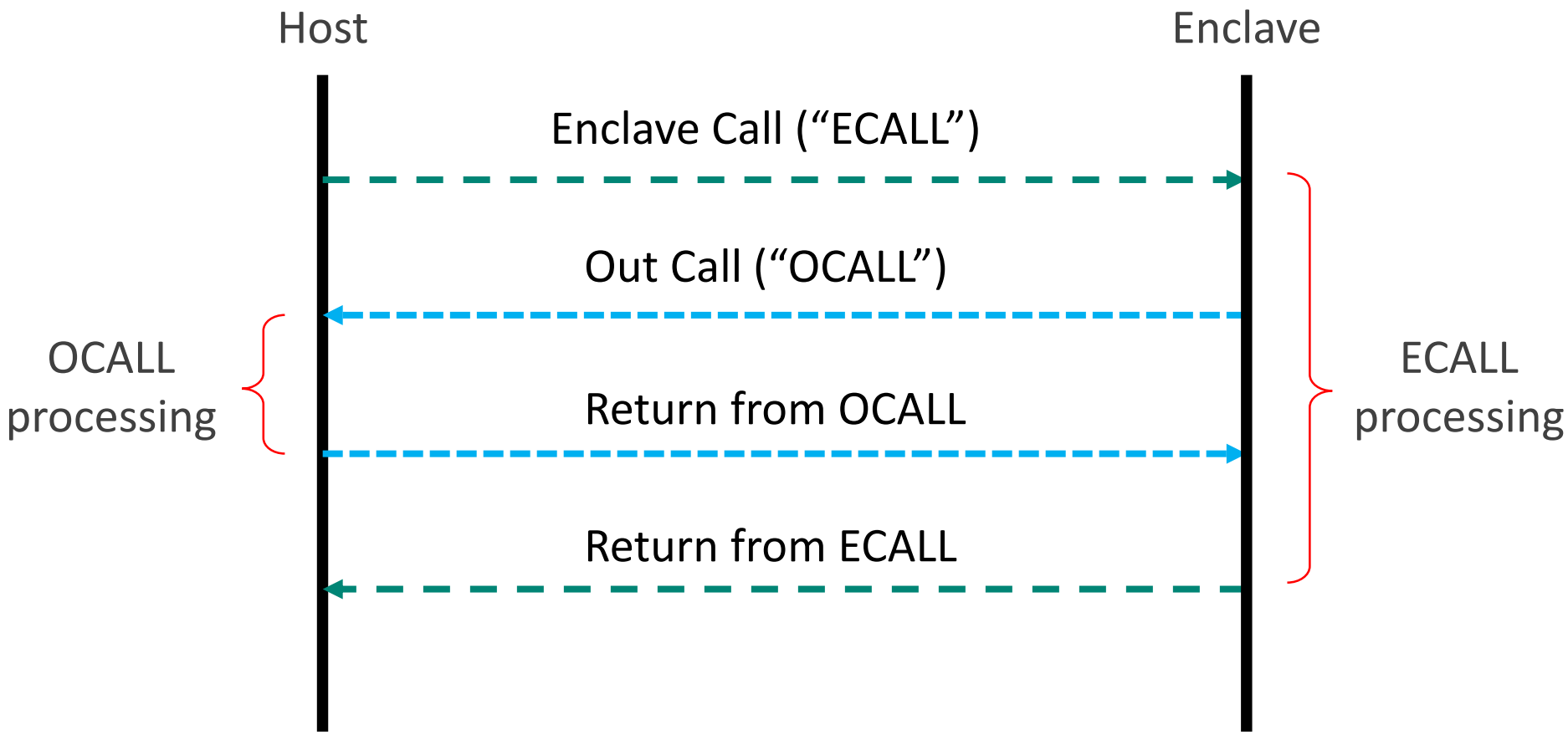
Open Enclave Application Architecture



Open Enclave Application Architecture

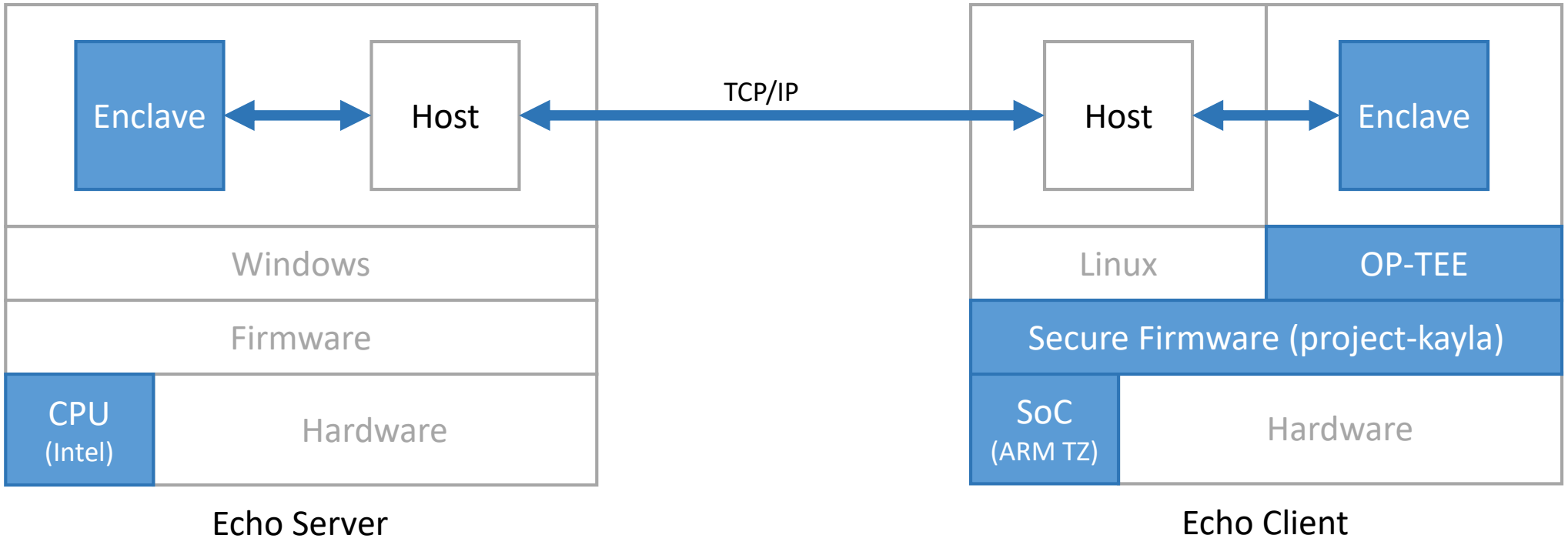


Enclave Calls Flow Diagram



Note: OP-TEE only supports 1 thread per TA, with ECALL holding a global lock

Example using “Echo” Sample



Trusted Component

What needs to be factored where?

Normal Application

- Loading of Trusted App
- Threading
- Timers
- Transport stack (e.g., sockets)
- Storage stack (if not provided by TEE)

Trusted Application

- Security credentials
- All operations using sensitive data
- I/O for trusted peripherals (if any)

Links

- https://github.com/Microsoft/openenclave
- Visual Studio extension: search for “Open Enclave”
- Visual Studio Code extension: search for Open Enclave
- Channel 9 episode: https://channel9.msdn.com/Shows/Internet-of-Things-Show/Deep-Dive-Confidential-Computing-in-IoT-using-Open-Enclave-SDK