Uncovering the Mysteries of Trusted Execution Environments: a Software System Perspective

Journées Sécurité - SiF

14 October 2021

Dr Valerio Schiavoni

valerio.schiavoni@unine.ch

University of Neuchâtel, Switzerland

Things we do not understand

Uncovering the Mysteries of Trusted Execution Environments: a Software System Perspective

Journées Sécurité - SiF

14 October 2021

Dr Valerio Schiavoni

valerio.schiavoni@unine.ch

University of Neuchâtel, Switzerland

Agenda

1. **Why** Trusted Execution Environments are important ?

2. **What** are TEEs after all ?

3. **When** to use or not to use TEEs ?

4. **Where** do we find TEEs nowadays ?

5. **How** to use TEEs?

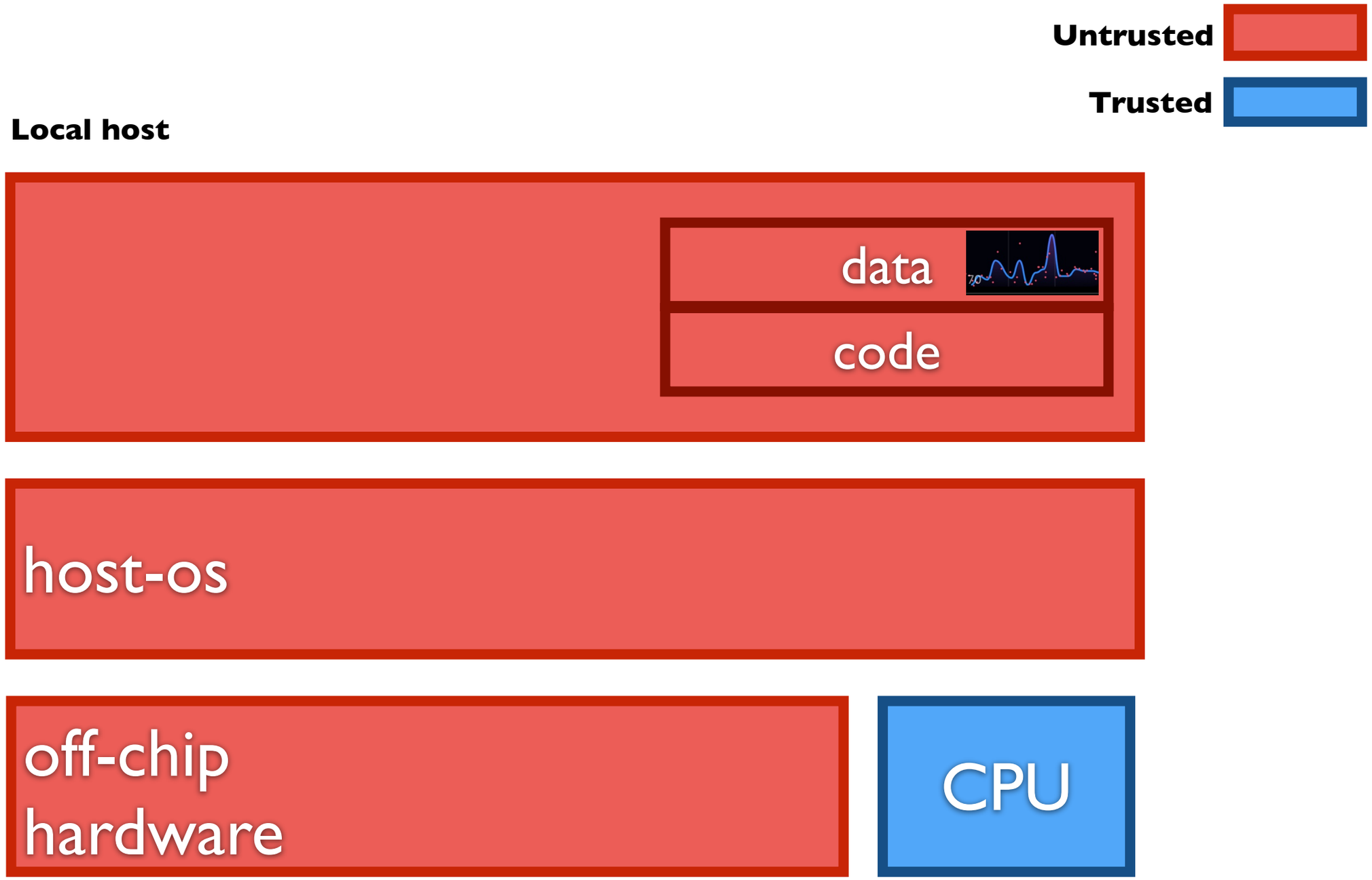
Motivating Scenario

- Suppose you want to develop an online service to handle very sensitive data
  - E.g., ECG logs
- Data **privacy** is paramount
  - Only for allowed stakeholders

Source: my heart

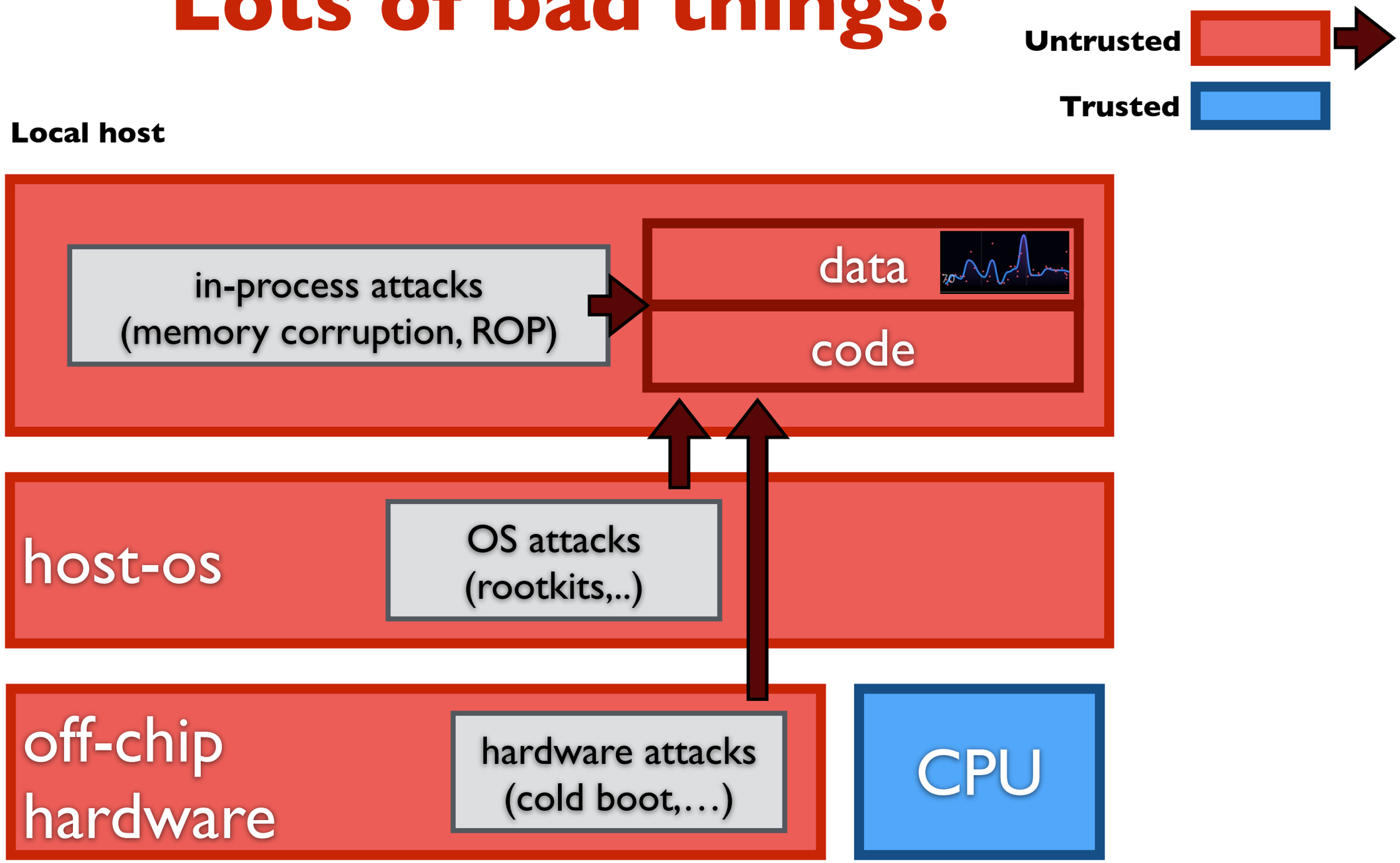
- Data **integrity** is paramount
  - If data integrity is compromised, risks of false alerts
- The **code** being executed must also be confidential
  - E.g., **algorithms** to compute HR variations and detect health anomalies

What could go wrong ?

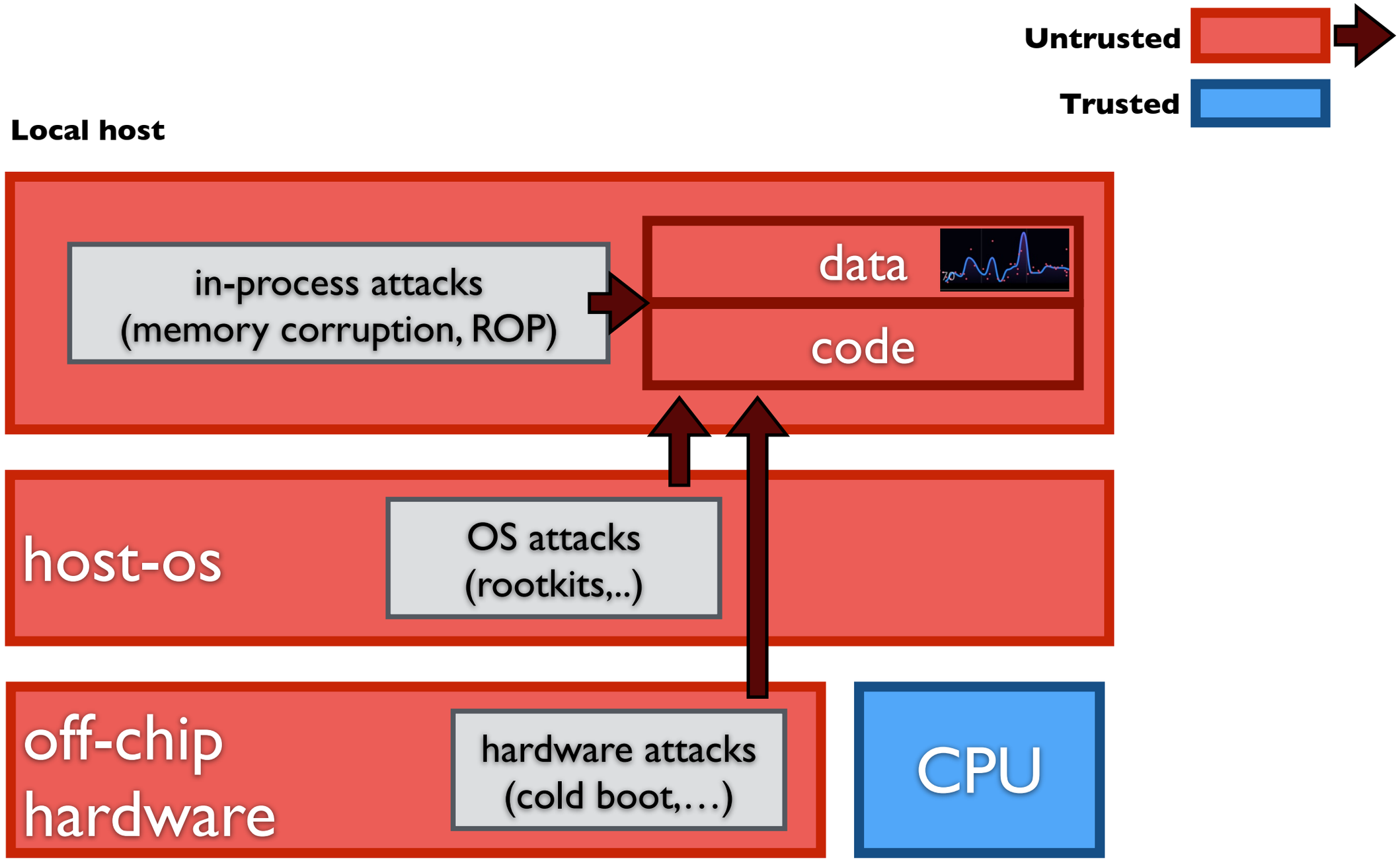


What could go wrong ?

Lots of bad things!

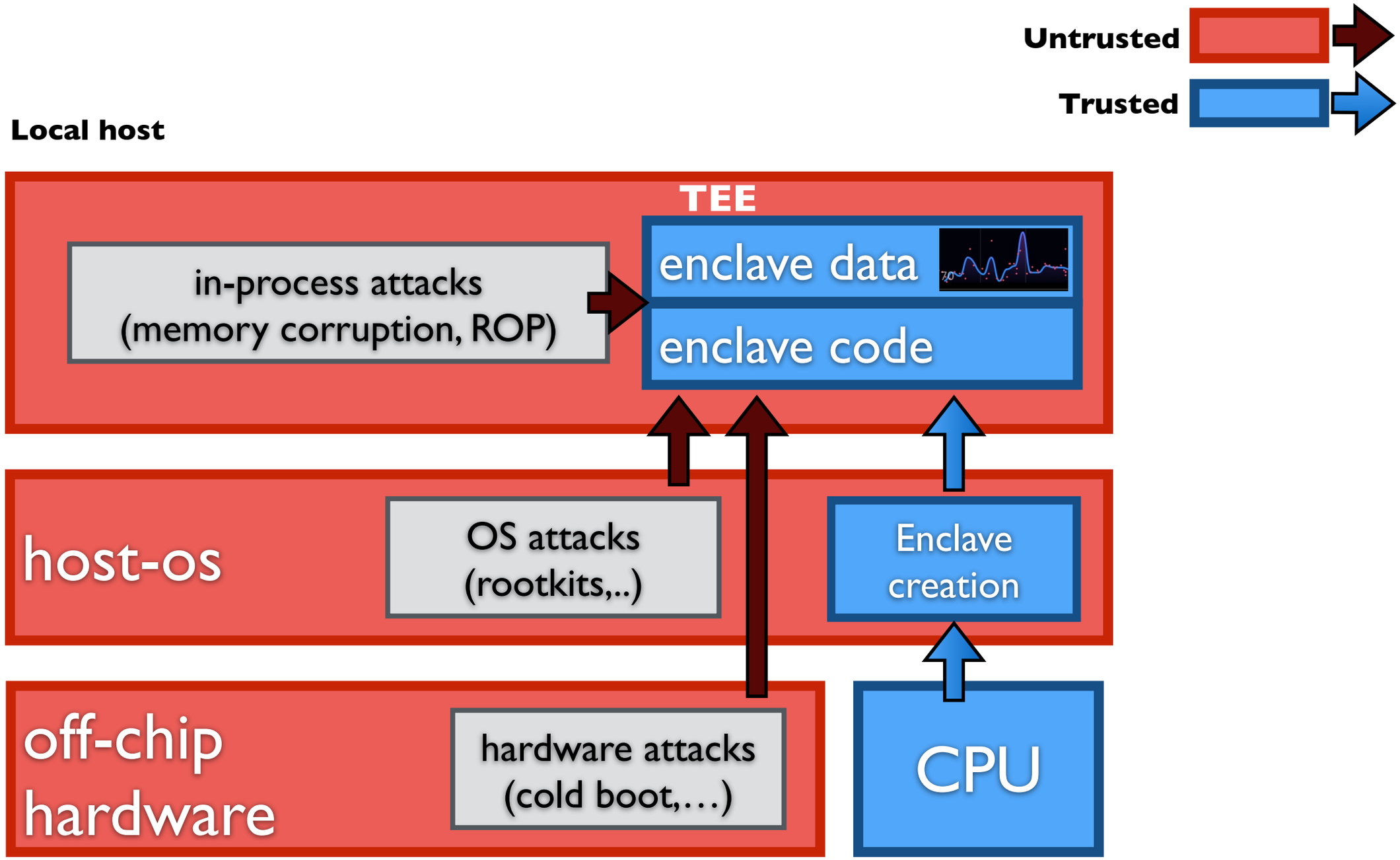


What could go wrong ?



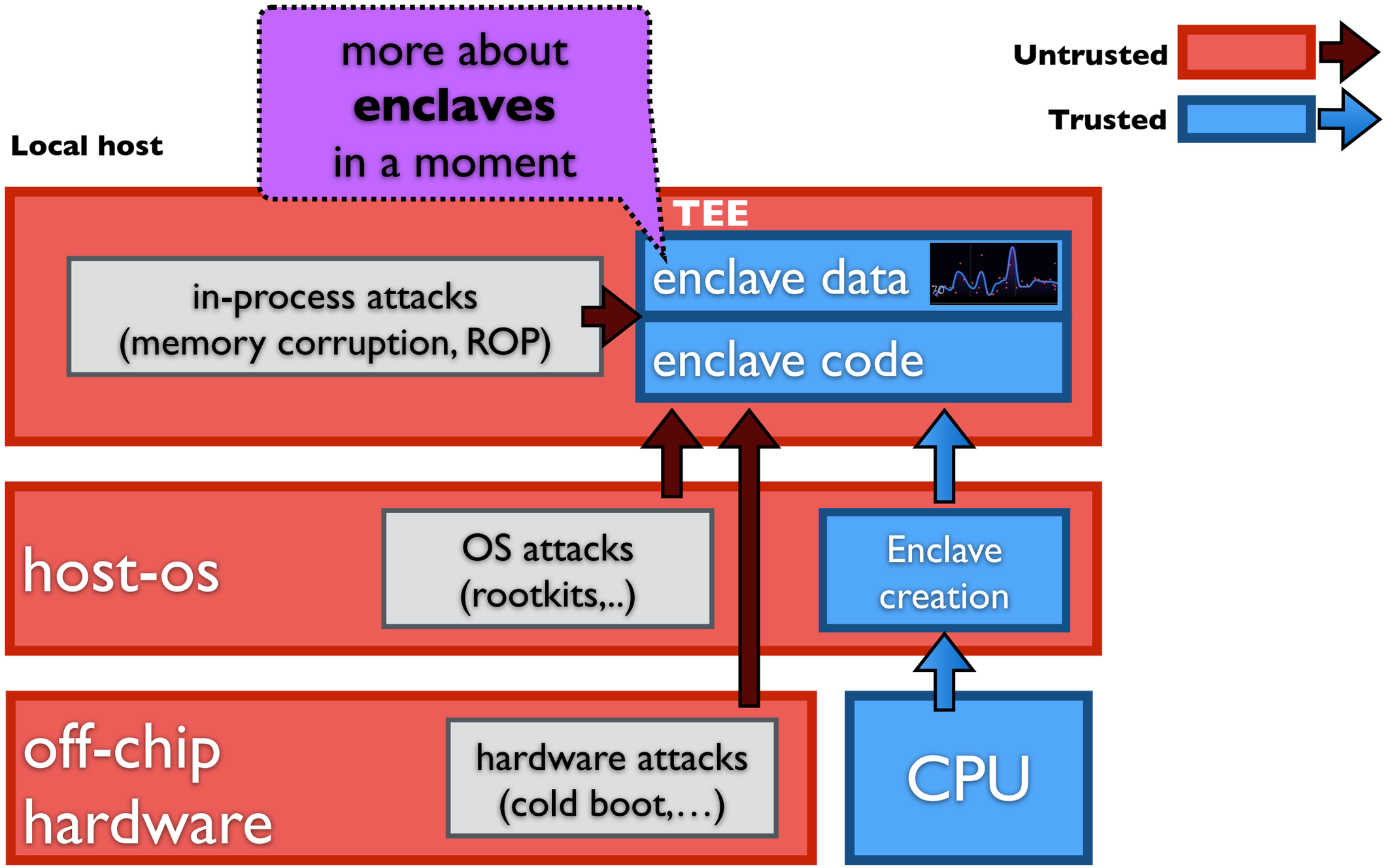
TEEs to the rescue !

What could go wrong ?



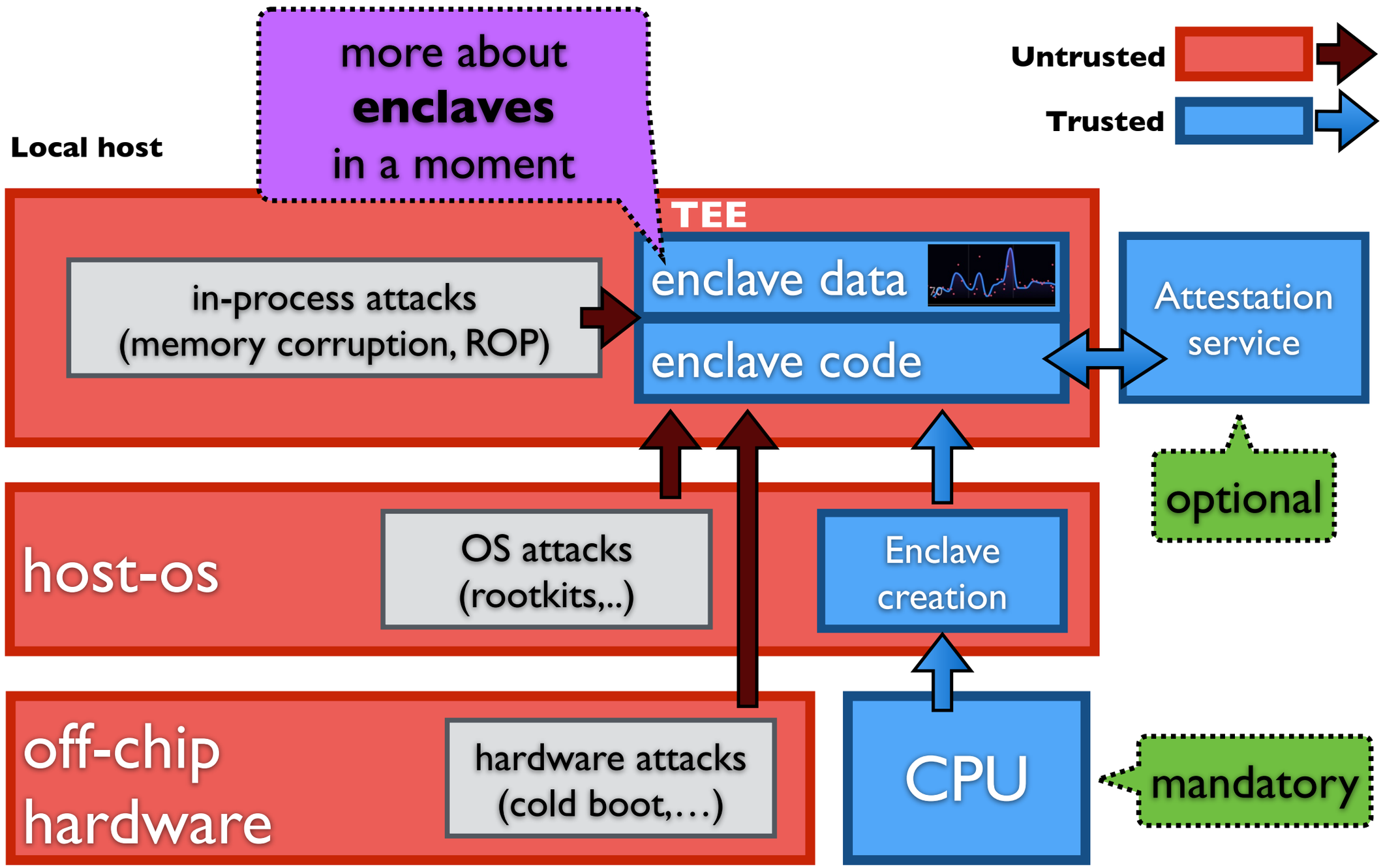
TEEs to the rescue !

What could go wrong ?



TEEs to the rescue !

What could go wrong ?



TEEs to the rescue !

Agenda

1. **Why** Trusted Execution Environments are important ?

   Protect code and data from powerful attackers

2. **What** are TEEs after all ?

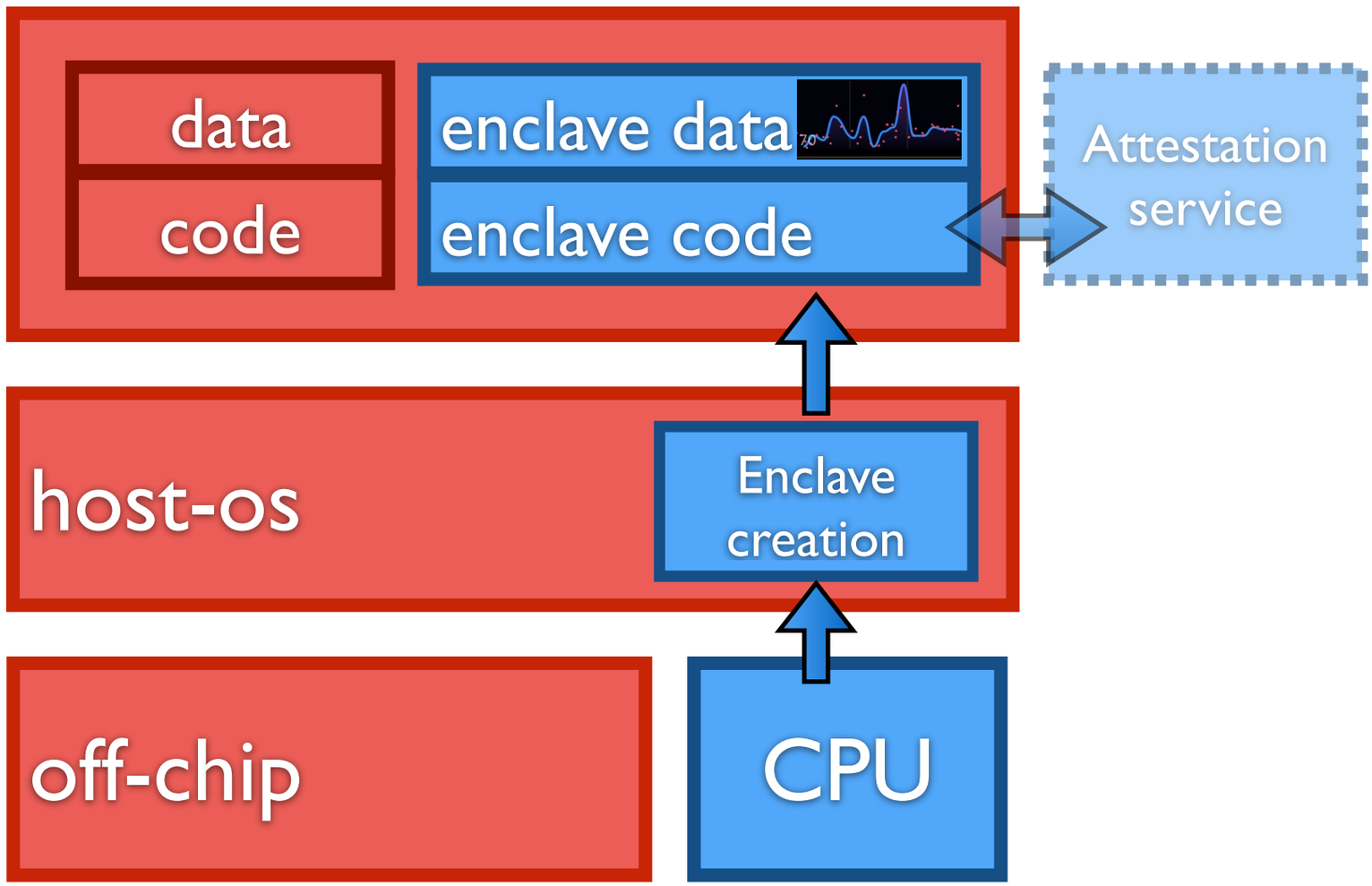
3. **When** to use or not to use TEEs ?

4. **Where** do we find TEEs nowadays ?

5. **How** to use TEEs?

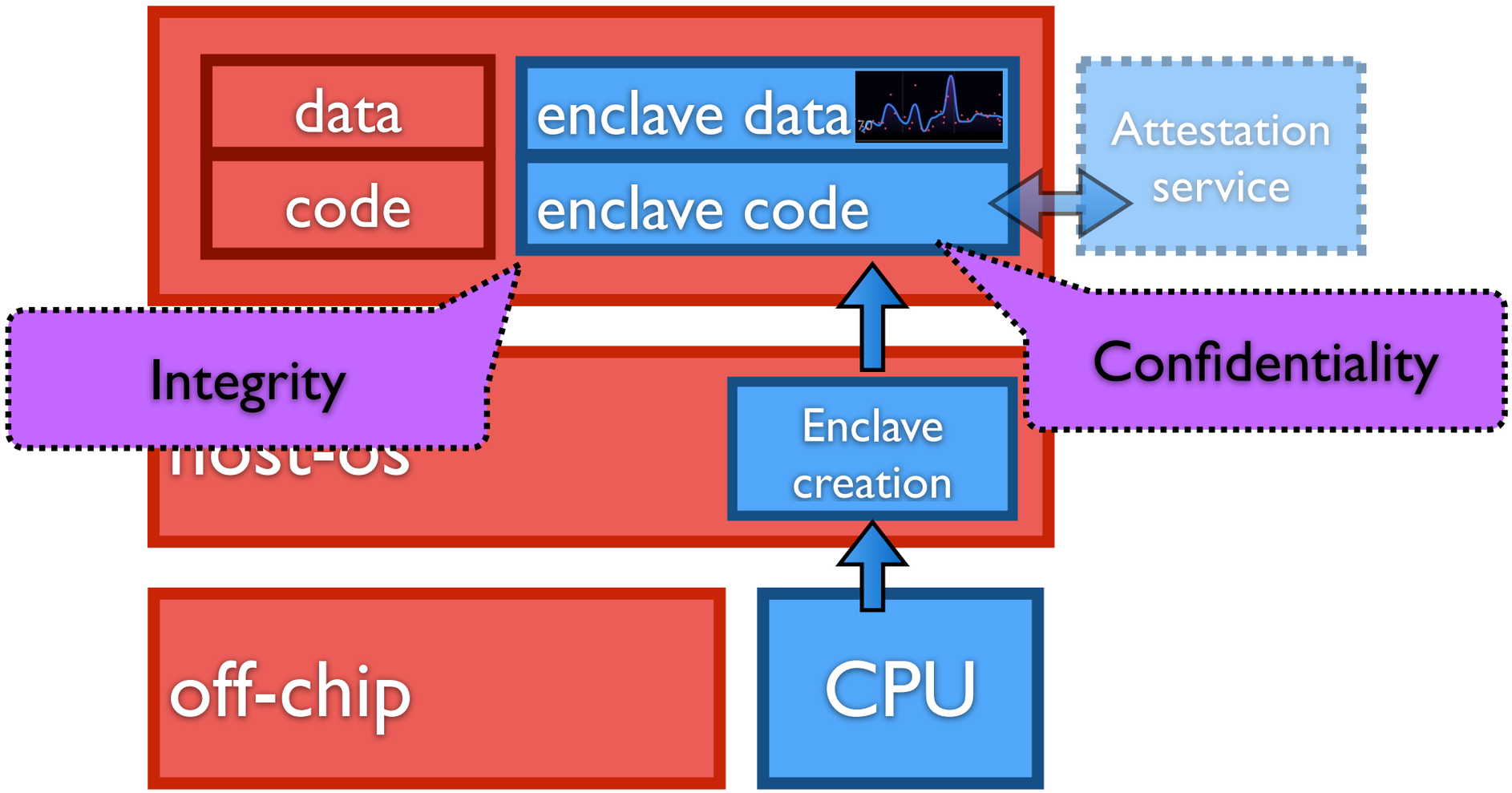
What is a TEE ?

- Hardware **protected** area against powerful attacks
- The **content** of the enclaves is **shielded** from:
  - Compromised operating system, compromised system libraries, attackers with physical access to a machine



What is a TEE ?

- Hardware **protected** area against powerful attacks
- The **content** of the enclaves is **shielded** from:
  - Compromised operating system, compromised system libraries, attackers with physical access to a machine



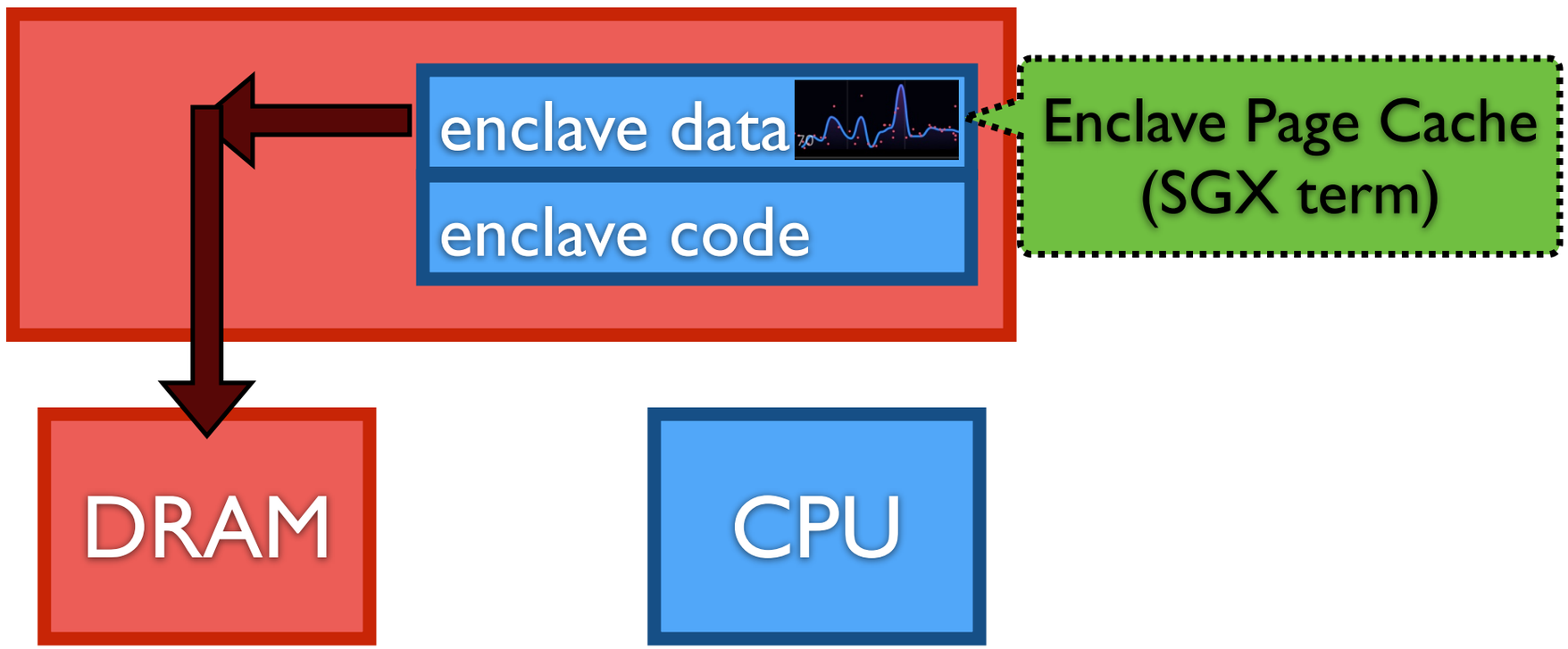
Confidentiality

- Code and data in the enclave never leave the CPU package unencrypted
  - ➡Outside the CPU, everything is encrypted

- When memory is read back into cache lines, the CPU decrypts

Confidentiality

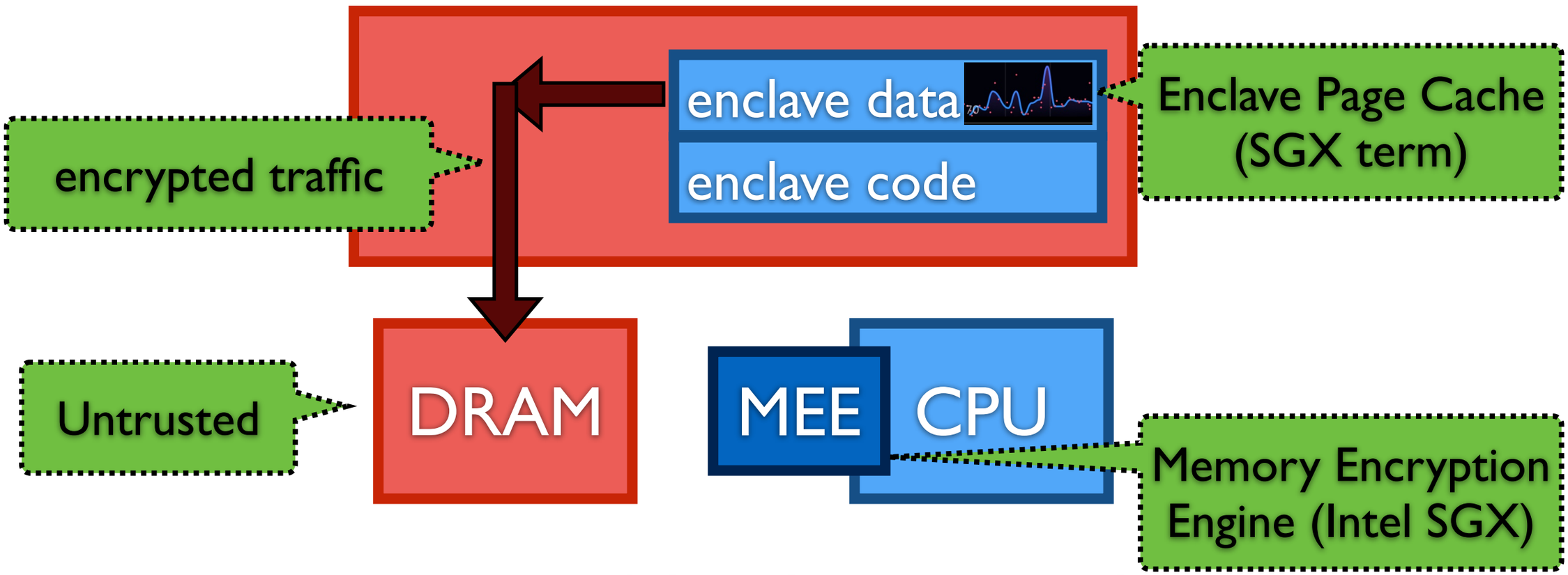
- Code and data in the enclave never leave the CPU package unencrypted
  - ➡Outside the CPU, everything is encrypted



- When memory is read back into cache lines, the CPU decrypts

Confidentiality

- Code and data in the enclave never leave the CPU package unencrypted
  - ➡Outside the CPU, everything is encrypted



- When memory is read back into cache lines, the CPU decrypts **(with the help of the MME)**

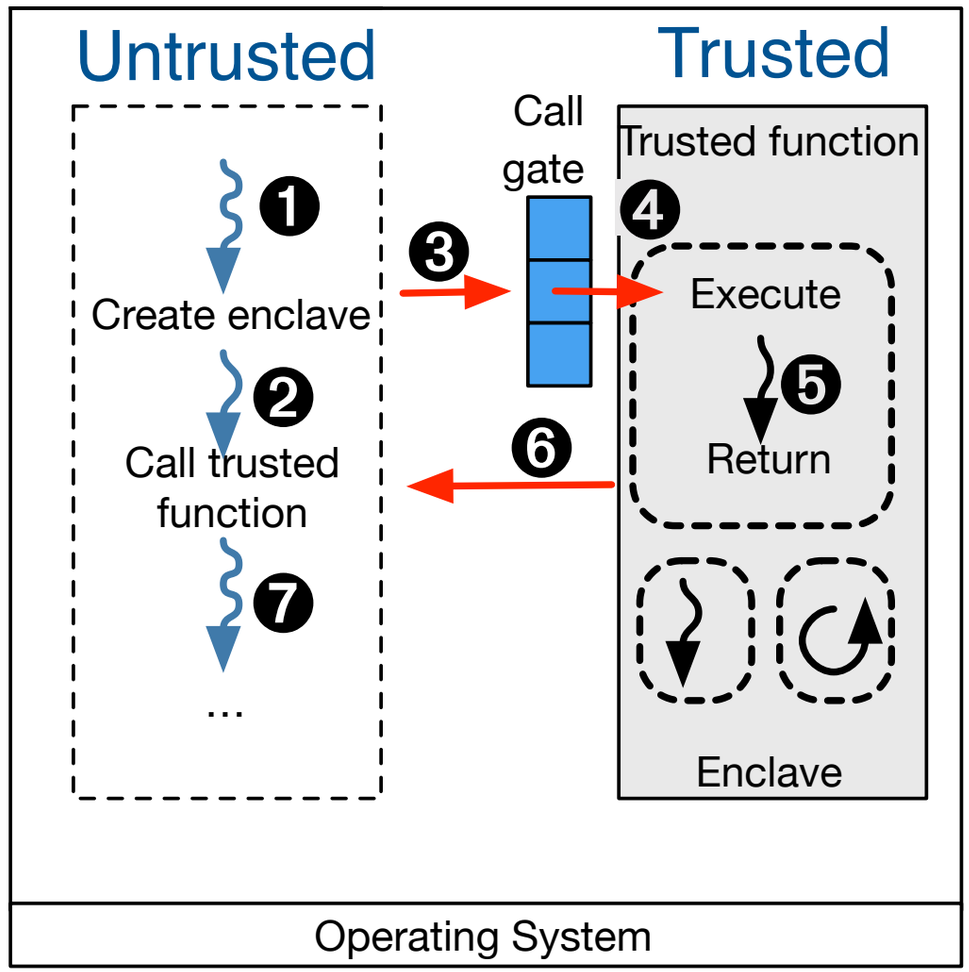
Integrity

CPU vendor-dependant by definition (see next)

- The **CPU** verify the **integrity** of cache lines
- The **CPU** verify the **integrity** of virtual-to-physical addresses
  - Intel SGX: MME maintains the root of a Merkle tree
  - Arm TrustZone: vendor-specific.
    - Example: Samsung's Knox uses passive and active counter-measures
  - In the case of AMD SEV: no integrity

Intel SGX in a nutshell

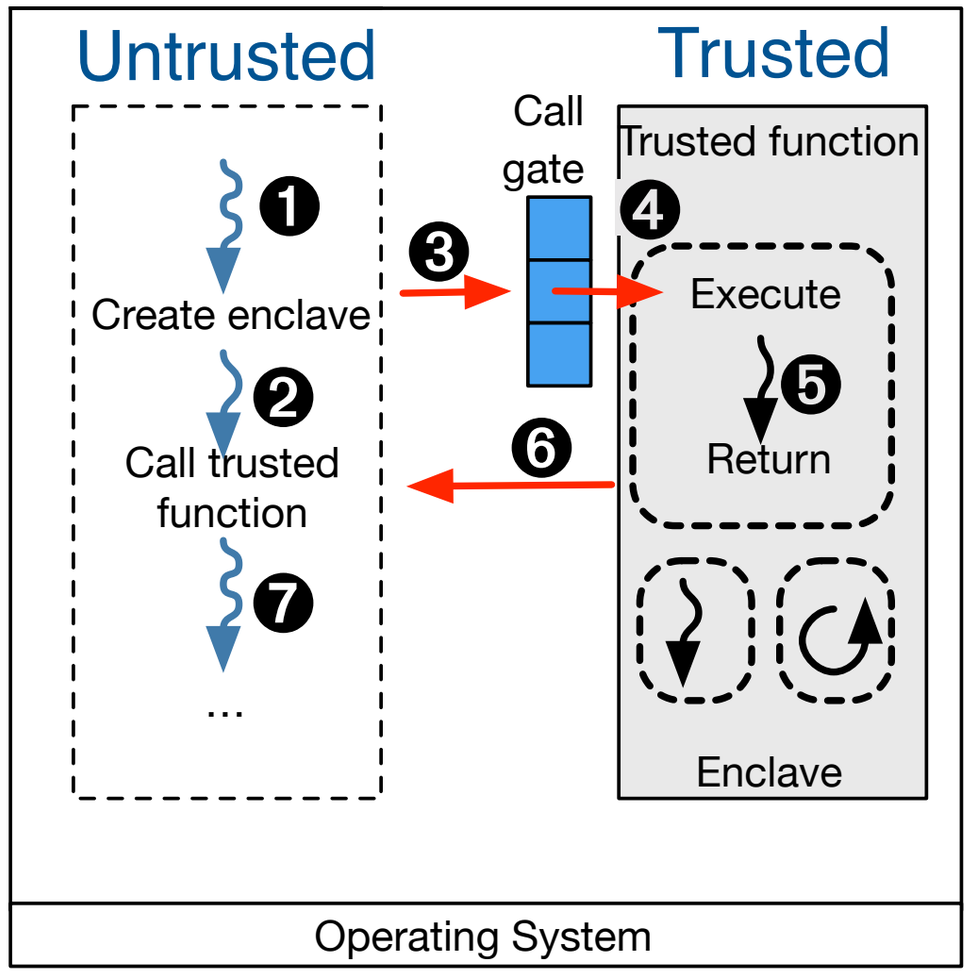
- Available since 2015, SkyLake
- Hardware-protected area on die
- Support strong adversarial models
- Split the program in two parts:
  - Untrusted vs. trusted, **enclaves**
- **Code integrity**, genuine hardware
- Intel Attestation Service
- **Memory limits**, EPC, up to 512 MB in recent server-grade processors, up to 128 MB until recently
- Intel SDK, C/C++, Rust SDK, frameworks for legacy systems (Scone, SGX-LKL, graphene-sgx, etc.)



Intel SGX in a nutshell

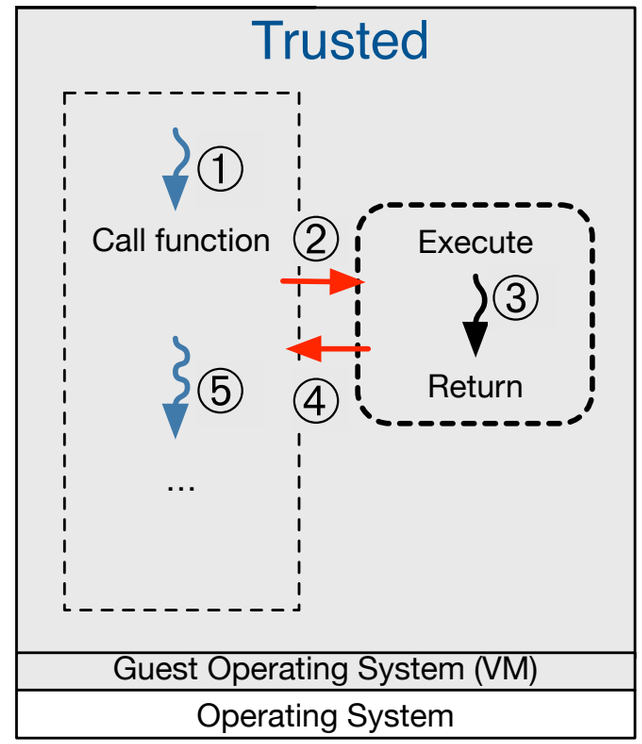
- ake
- n die
- Support strong adversarial models
- Split the program in two parts:
  - Untrusted vs. trusted, **enclaves**
- **Code integrity**, genuine hardware
- Intel Attestation Service
- **Memory limits**, EPC, up to 512 MB in recent server-grade processors, up to 128 MB until recently
- Intel SDK, C/C++, Rust SDK, frameworks for legacy systems (Scone, SGX-LKL, graphene-sgx, etc.)

How to split is non-trivial.
Manual? Automatic? No-split?
It also depends on the language



AMD SEV in a nutshell

- **S**ecure **E**ncrypted **V**irtualization
- Secure Memory Encryption
- Designed for virtualized systems
- Lack of integrity protection
  - SEV-SNMP fixing this
- Attestation
  - To fix in hardware?



SEVered: Subverting AMD's Virtual Machine Encryption

Mathias Morbitzer, Manuel Huber, Julian Horsch and Sascha Wessel
Fraunhofer AISEC
Garching near Munich, Germany
{firstname.lastname}@aisec.fraunhofer.de

EuroSec'18

Insecure Until Proven Updated: Analyzing AMD SEV's Remote Attestation

Robert Buhren
robert.buhren@sect.tu-berlin.de
Technische Universität Berlin
Security in Telecommunications

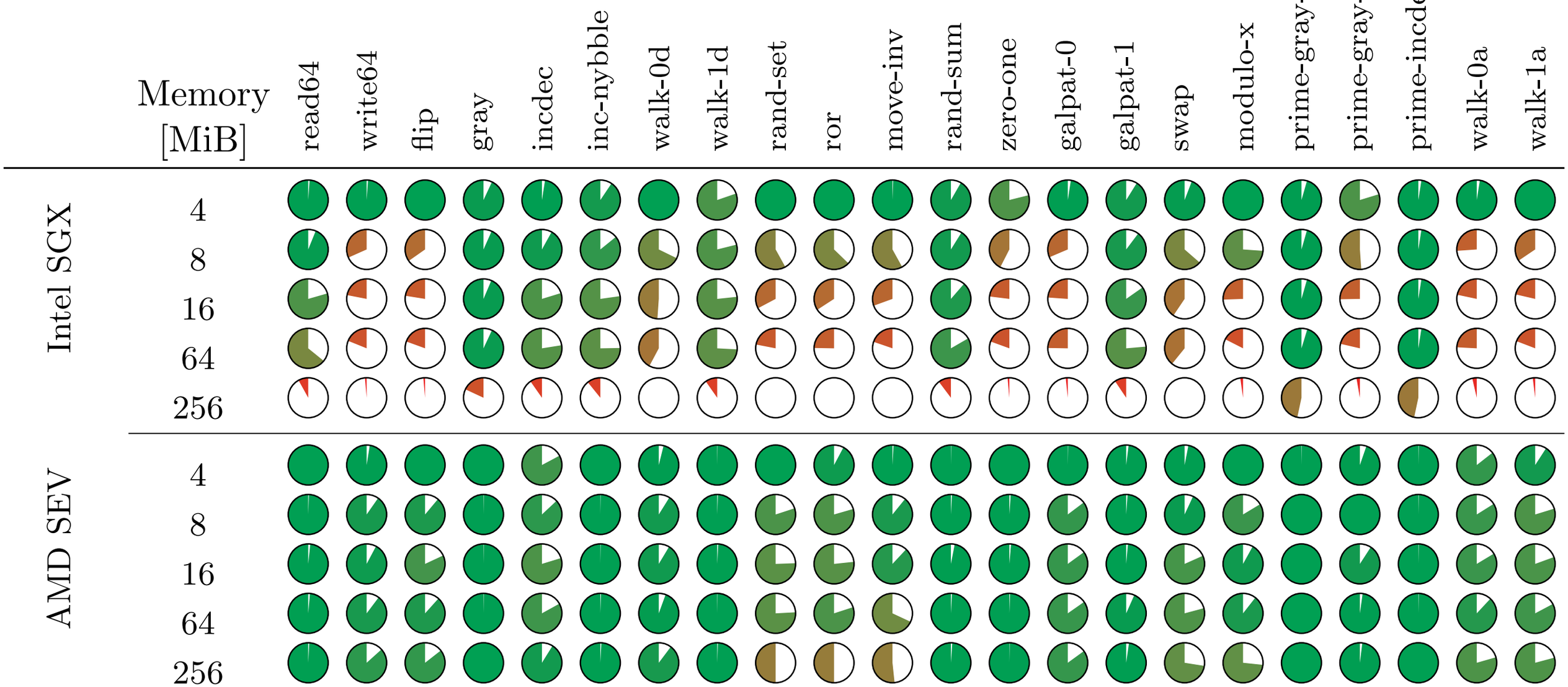
Christian Werling
christian.werling@student.hpi.de
Hasso Plattner Institute, Potsdam

Jean-Pierre Seifert
jpseifert@sect.tu-berlin.de
Technische Universität Berl
Security in Telecommunicati

CCS'19

SGX vs. SEV

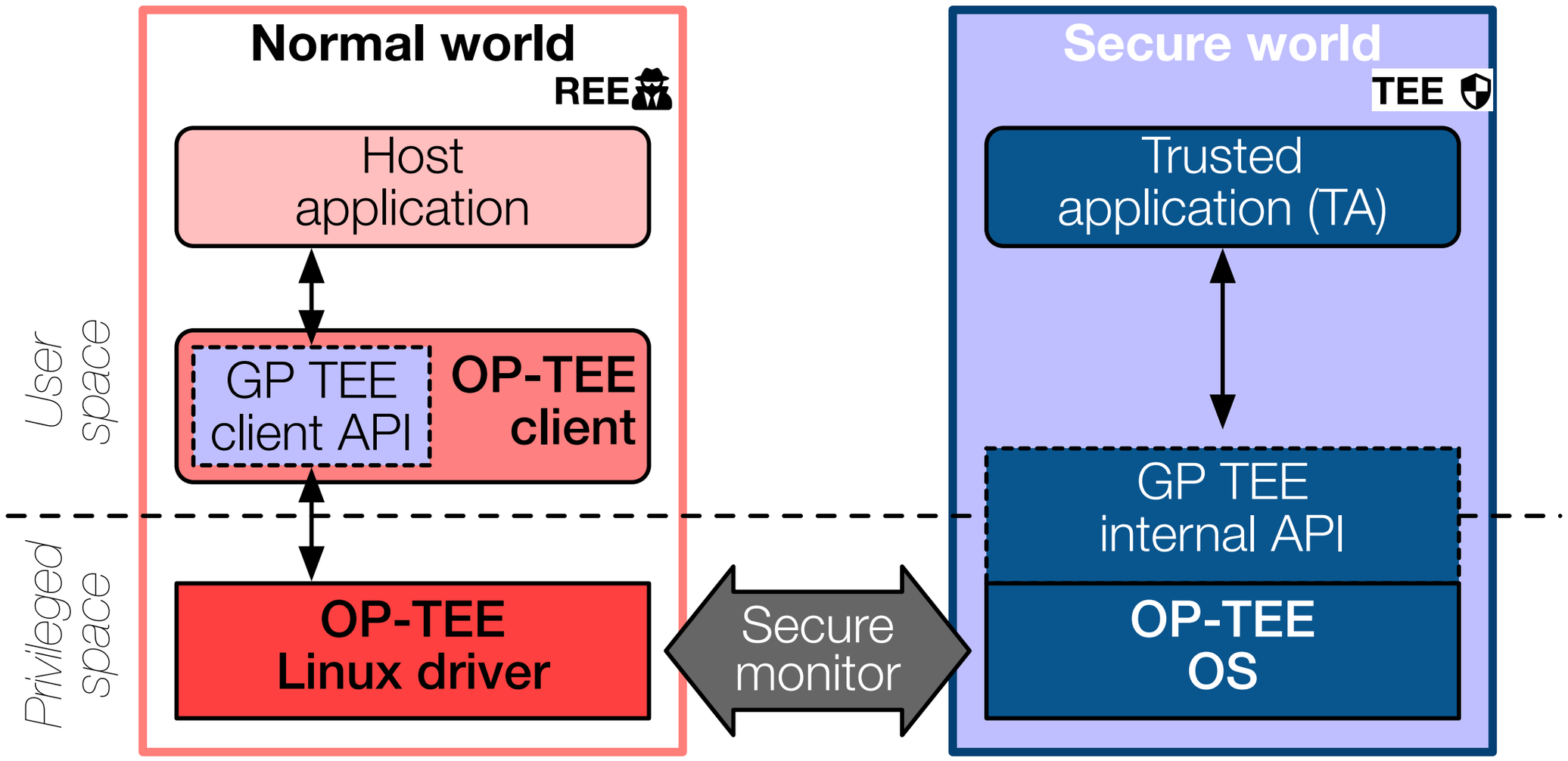
- Memory-bound stressors from `stress-ng`
- The **greener**, the better



TrustZone in a nutshell

- Two-world separation, one TA at the time
- Lack of built-in attestation service
- 2~5Mb per TA

Watch upcoming talk on OP-TEE for more !



Other TEEs ?

- Risc-V:
  - MultiZone
  - KeyStone
  - Penglai

Discussed later today

- Since 2017, Google's Titan M on Android Pixel (since v3)
- IBM SecureBlue & SecureBlue++
- Upcoming new ARM Confidential Compute Architecture (CCA)

*(or, the **main** questions to answered by this presentation)*

Agenda

1. **Why** Trusted Execution Environments are important ?

   Protect code and data from powerful attackers

2. **What** are TEEs after all ?

   HW-shielded areas to build stronger systems

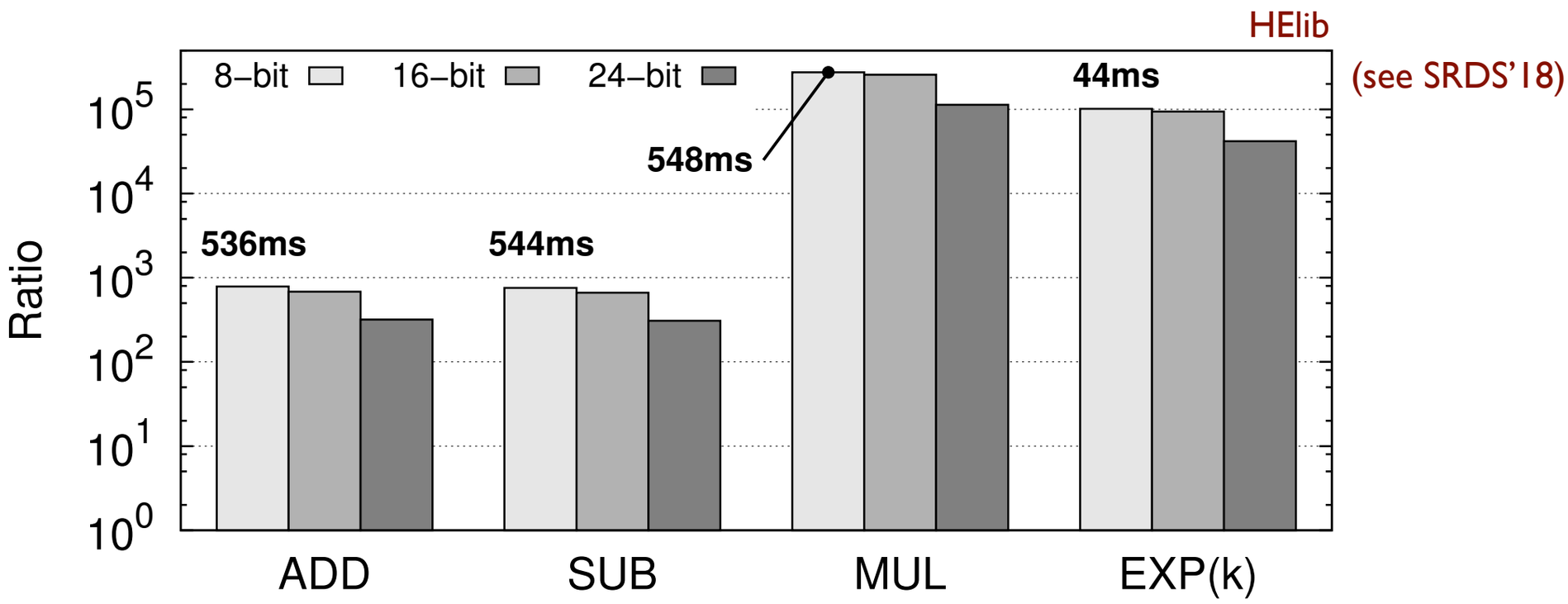
3. **When** to use or not to use TEEs ?

4. **Where** do we find TEEs nowadays ?

5. **How** to use TEEs?

Why TEEs are good?

- Operations **inside** TEEs run at bare-metal speed
- Strong adversarial models (i.e., compromised OS)
- **Orders of magnitude** faster than SotA homomorphic encryption



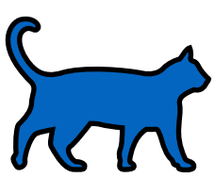
- Microsoft SEAL
- Google Private Join and Compute?

Why TEEs are not-so-good?

- At least in the current incarnations:
  1. Requires some **craft** from programmers
  2. Might lack fundamental properties
  3. Performances can be poor (goto 1)
  4. Requires good knowledge of system issues
  5. Continuous stream of side-channel attacks (and fancy logos)!

Intel won't fix (outside threat model of SGX)

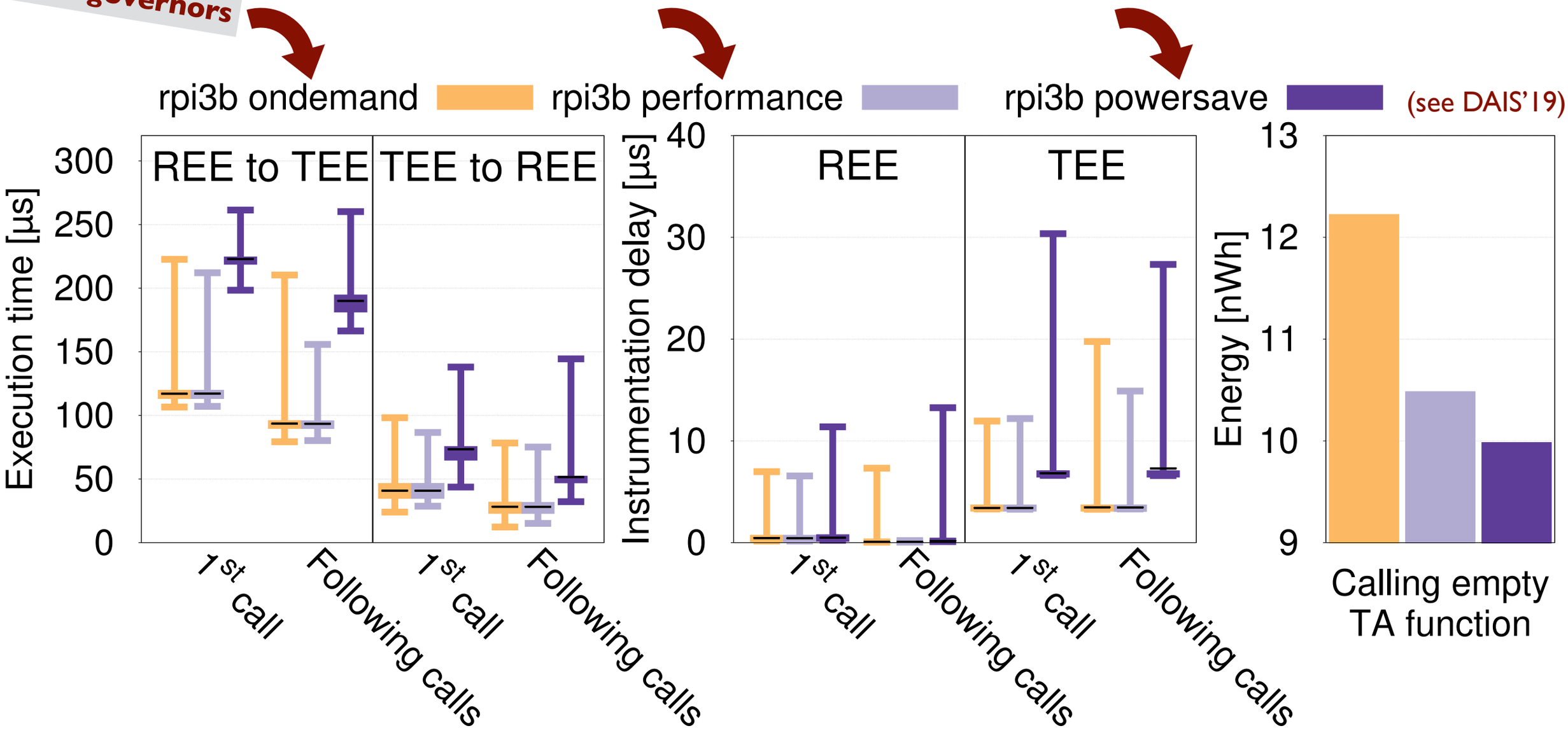
- Followed by a stream of mitigations, patches..



Can target several TEEs

TZ: Context Switching

3 CPU governors



- From REE (untrusted) to TEE (trusted) on average more expensive
- **Energy** side-effects, check CPU governors

Devs must know system details

*(or, the **main** questions to answered by this presentation)*

Agenda

1. **Why** Trusted Execution Environments are important ?

   Protect code and data from powerful attackers

2. **What** are TEEs after all ?

   HW-shielded areas to build stronger systems

3. **When** to use or not to use TEEs ? **Not so easy!**

   Requires awareness on the target scenarios

4. **Where** do we find TEEs nowadays ?

5. **How** to use TEEs?

TEEs at the edge

- **Mobile** devices based on Arm processors might support TrustZone
  - Samsung Knox TEE
  - Google Trusty
- Cheap off-the-shelf **IoT devices** with Arm processors
  - Raspberry 3, 3B+, 4
- Intel compute-sticks (size of an USB stick)

Server-grade TEEs

- Intel SGX widely available on server-grade processors
  - EPC limits being lifted
    - With Total Memory Encryption (TME) and MKTME (multi-key), EPC might become obsolete
- AMD processors largely support SEV
  - Next iteration will ship integrity

Cloud Providers

#TEE-as-a-Service#

- **AWS TEE-enabled instances**
  - AWS Nitro Enclaves. Several limits: no persistent storage, no network, no human access
  - Communication via a trusted channel with the instance creating the nitro enclave
  - (speculation) Some flavour of TrustZone
- **Azure Confidential Computing instances**
  - Intel-SGX based VM
  - Support for attestation services (both Azure and Intel)
  - Open Enclave https://openenclave.io/sdk/
- **Google Shielded VMs**
  - Secure boot, vTPM, integrity monitoring
  - https://cloud.google.com/shielded-vm/

*(or, the **main** questions to answered by this presentation)*

Agenda

1. **Why** Trusted Execution Environments are important ?

   Protect code and data from powerful attackers

2. **What** are TEEs after all ?

   HW-shielded areas to build stronger systems

3. **When** to use or not to use TEEs ? **Not so easy!**

   Requires awareness on the target scenarios

4. **Where** do we find TEEs nowadays ?

   Edge, Cloud, and everything in between

5. **How** to use TEEs?

How-to TEEs

- If you reach this point, you have solved already the majority of the **mysteries** surrounding TEEs (why, what, when, where)!
  - Well done, the hard part is beyond you !
- The next (big) task is common in the software-lifecycle:
  - design
  - implement
  - test
  - deploy
  - profile
  - debug

How-to TEEs

- If you reach this point, you have solved already the majority of the **mysteries** surrounding TEEs (why, what, when, where)
- The next (big) tasks are common in the software-lifecycle:
  - ~~design~~
  - **implement**
  - ~~test~~
  - **deploy**
  - **profile**
  - ~~debug~~

What are the specific difficulties in executing these steps when dealing with TEEs ?

will refer to SGX examples

How-to implement TEEs

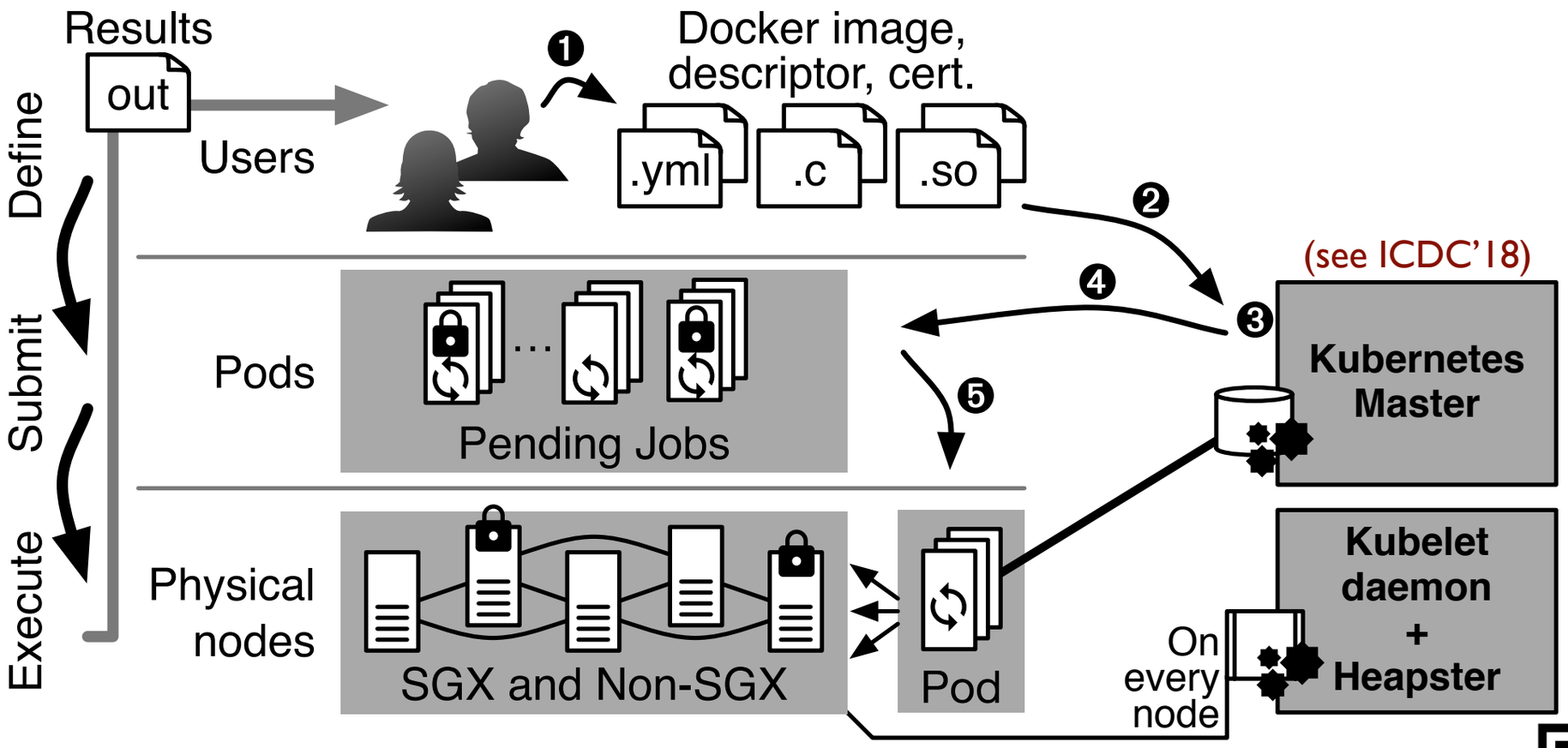
- Dealing with legacy-software:
  - Limited choice of implementation languages
  - Rely on intermediate representations:
    - WebAssembly (check-out our **Twine** system, ICDE'21)
    - Java (check-out our **Montsalvat**, Middleware'21)
- If you are lucky and work on a new TEE-enabled project:
  - Start from scratch, native C/C++ SDKs, Rust
  - Key decision: split between **untrusted**/**trusted**
  - Important **performance** consequences

open-source, contact me

Does it have to be this way? No…but at which cost?

How-to deploy TEEs

- Suppose you ended up with a container-enabled solution
  - How to schedule programs in TEE-enabled clusters?
  - Prototype extends Kubernetes and Intel SGX driver



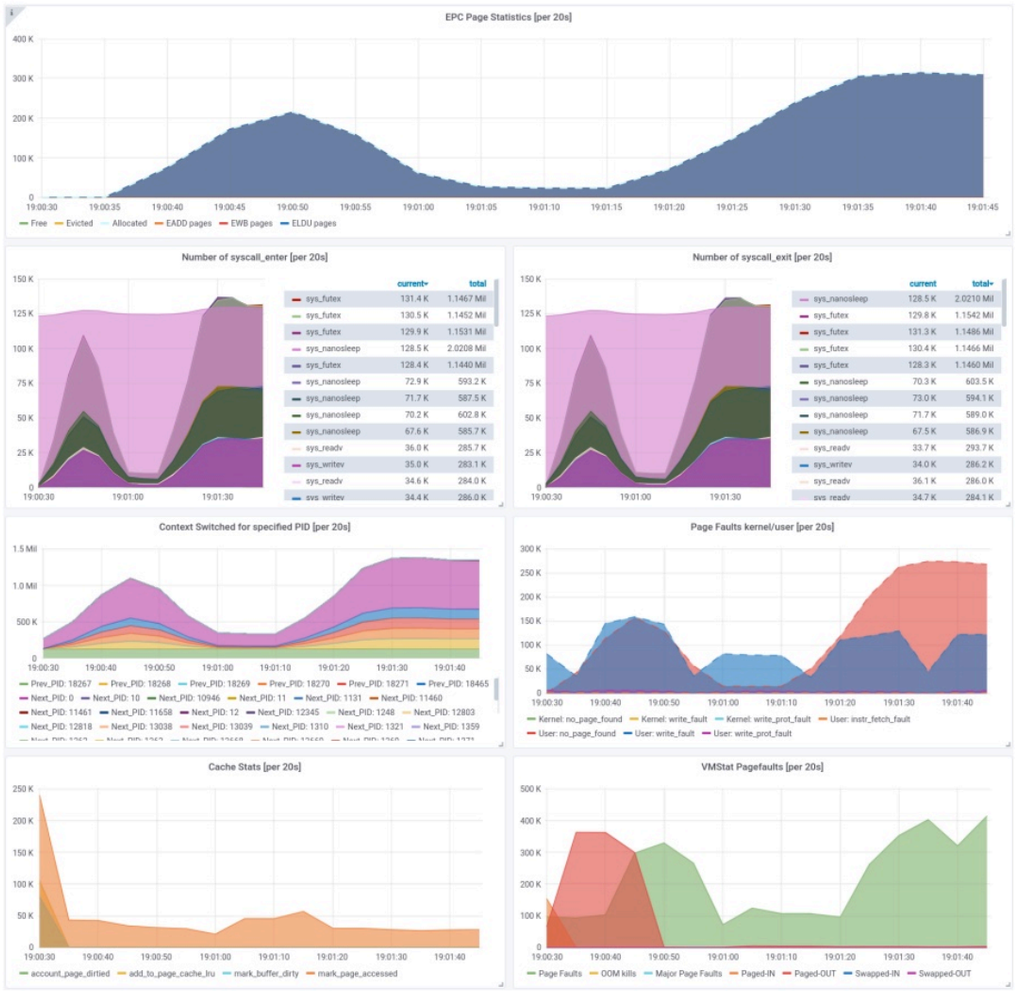
- Open-source: https://github.com/sebva/sgx-orchestrator

How-to profile TEEs

- You might find out that your TEE-powered program is slow
- How do you profile it ?
- Requires specialised tools

Tool	Framework Agnostic	Paging	Enclave Transitions	Orchestrated Applications	Real-Time Reports
LIKWID	✓	✗	✗	✓	✗
perf	✓	✗	✗	✗	✗
MemProf	✓	✗	✗	✗	✗
TEE-Perf	✓	✗	✗	✗	✗
gprof	✓	✗	✗	✗	✗
VTune	✓	✗	✗	✗	✗
SGX-Perf	✗	✓	✓	✗	✗
SGXTOP	✓	✓	✓	✗	✓
TEEMon	✓	✓	✓	✓	✓

(see Middleware'20)



TEEMon

*(or, the **main** questions to answered by this presentation)*

Agenda

1. **Why** Trusted Execution Environments are important ?

   Protect code and data from powerful attackers

2. **What** are TEEs after all ?

   HW-shielded areas to build stronger systems

3. **When** to use or not to use TEEs ? **Not so easy!**

   Requires awareness on the target scenarios

4. **Where** do we find TEEs nowadays ?

   Edge, Cloud, and everything in between

5. **How** to use TEEs?

   Understand your requirements

Mysteries Solved

1. **Why** Trusted Execution Environments are important ?

   Protect code and data from powerful attackers

2. **What** are TEEs after all ?

   HW-shielded areas to build stronger systems

3. **When** to use or not to use TEEs ? **Not so easy!**

   Requires awareness on the target scenarios

4. **Where** do we find TEEs nowadays ?

   Edge, Cloud, and everything in between

5. **How** to use TEEs?

   Understand your requirements

Mysteries Solved

Any questions (or mysteries) left ?

1. **Why** Trusted Execution Environments are important ?

   Protect code and data from powerful attackers

2. **What** are TEEs after all ?

   HW-shielded areas to build stronger systems

3. **When** to use or not to use TEEs ? **Not so easy!**

   Requires awareness on the target scenarios

4. **Where** do we find TEEs nowadays ?

   Edge, Cloud, and everything in between

5. **How** to use TEEs?

   Understand your requirements

One Open Challenge

- **Mixed TEEs** communication
  - In heterogeneous deployments, it is probably the norm
  - How to build trusted channels between SGX and TZ enclaves?
  - How about the next ones (RISC-V Keystone,…)
  - What new systems can we build out of these primitives?