Confidential Computing

SGX Local Attestation

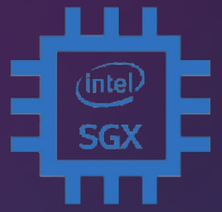
Prof. Dr. Christof Fetzer

Questions:

What is an enclave?

How can we create an enclave?

How can we trust an enclave if we cannot trust the server/OS/… ?



Intel SGX

- Deep Dive - introduce some technical concepts -

Intel SGX

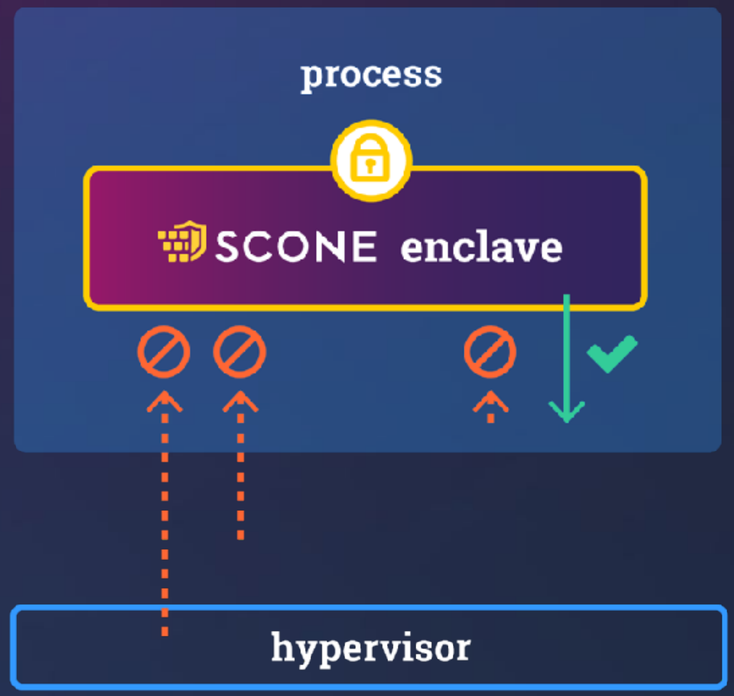
- SGX = “Software Guard eXtensions”

- CPU extension - now mainly in server CPUs
  - Skylake (2015), Kaby lake (2016), Coffee Lake (2017)
  - Xeon CPUs (Dec. 2018, Coffee Lake), Server CPUs (2021: Icelake, 2023: Saphire Rapids)

- Protects confidentiality & integrity of data and integrity of code in untrusted environments
  - Platform owner/root user is considered malicious
  - Access to main memory via management interface (BMC)
  - Only the **CPU chip** and the code in ***encrypted region*** (=enclave) are trusted

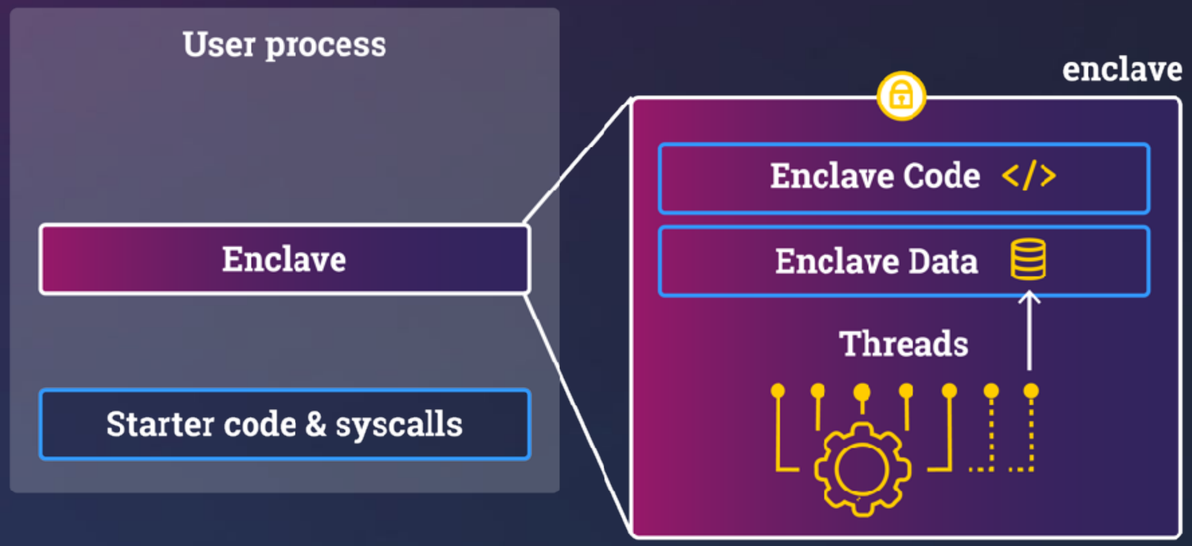
Enclave memory

- Enclave memory is encrypted and integrity-protected by the hardware
  - Memory encryption engine (MEE)
  - No plaintext secrets in the main memory
- Code in enclave can access outside memory
  - but outside cannot access memory in enclave



SCONE execution model

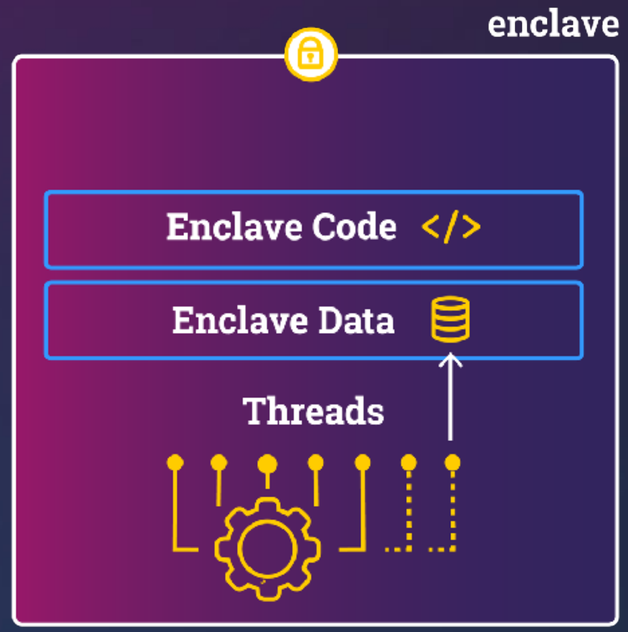
- Trusted execution environment in a process
- All application data & code inside of enclave
- Outside of enclave:
  - code to start enclave
  - code to execute system calls (via FIFO queue)



Enclave: Initial State is known!

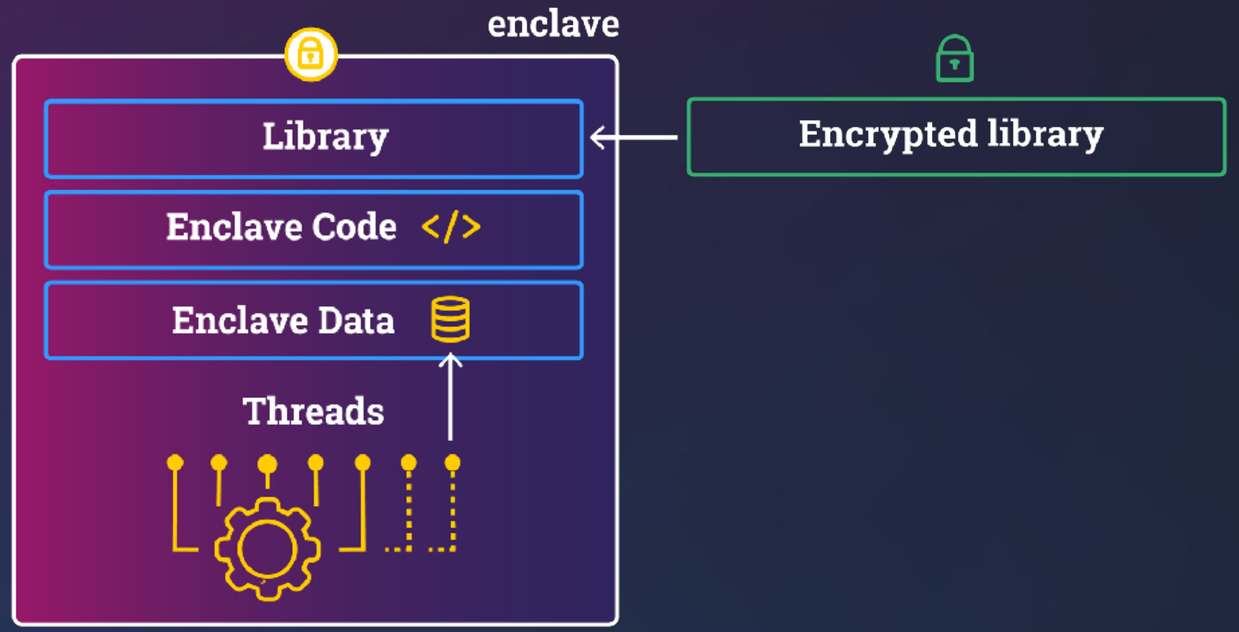
- When enclave is created,
  all enclave content is known!

Initial Enclave State
- no secrets!



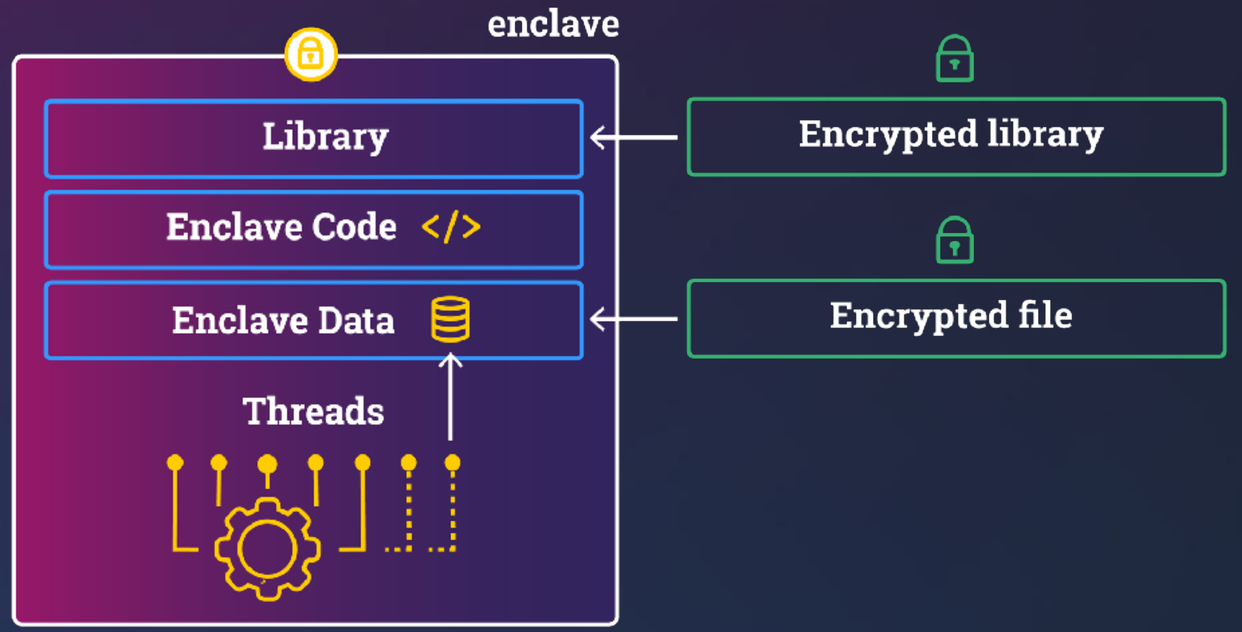
SCONE - Encrypted Code

- When enclave is created, all enclave content is known!
- After enclave is started, one can load encrypted libraries
  - which are transparently decrypted



SCONE - encrypted code

- When enclave is created, all enclave content is known!
- After enclave is started, one can load encrypted libraries
  - which are transparently decrypted
- Also, encrypted files can be loaded and are transparently decrypted

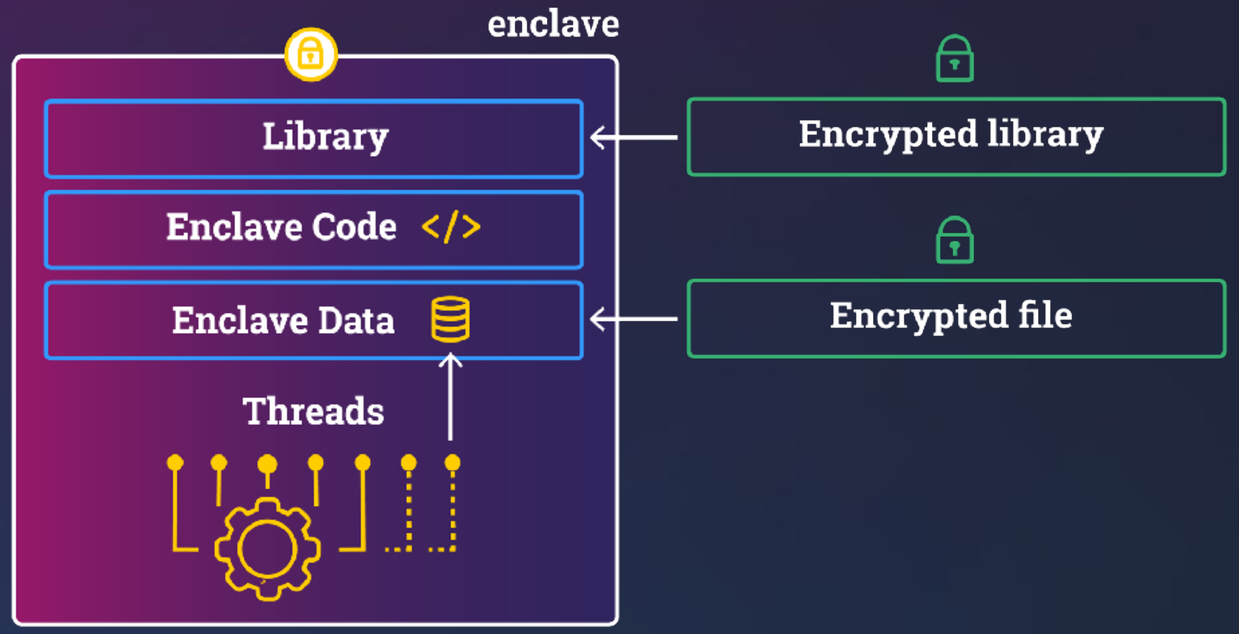


Initial Enclave State
- all data known -

Problem: Encryption Key

- Enclave does not know any secrets
- How can enclave decrypt the libraries / files?

Initially, no key known!



Approach: attestation & secret provisioning!

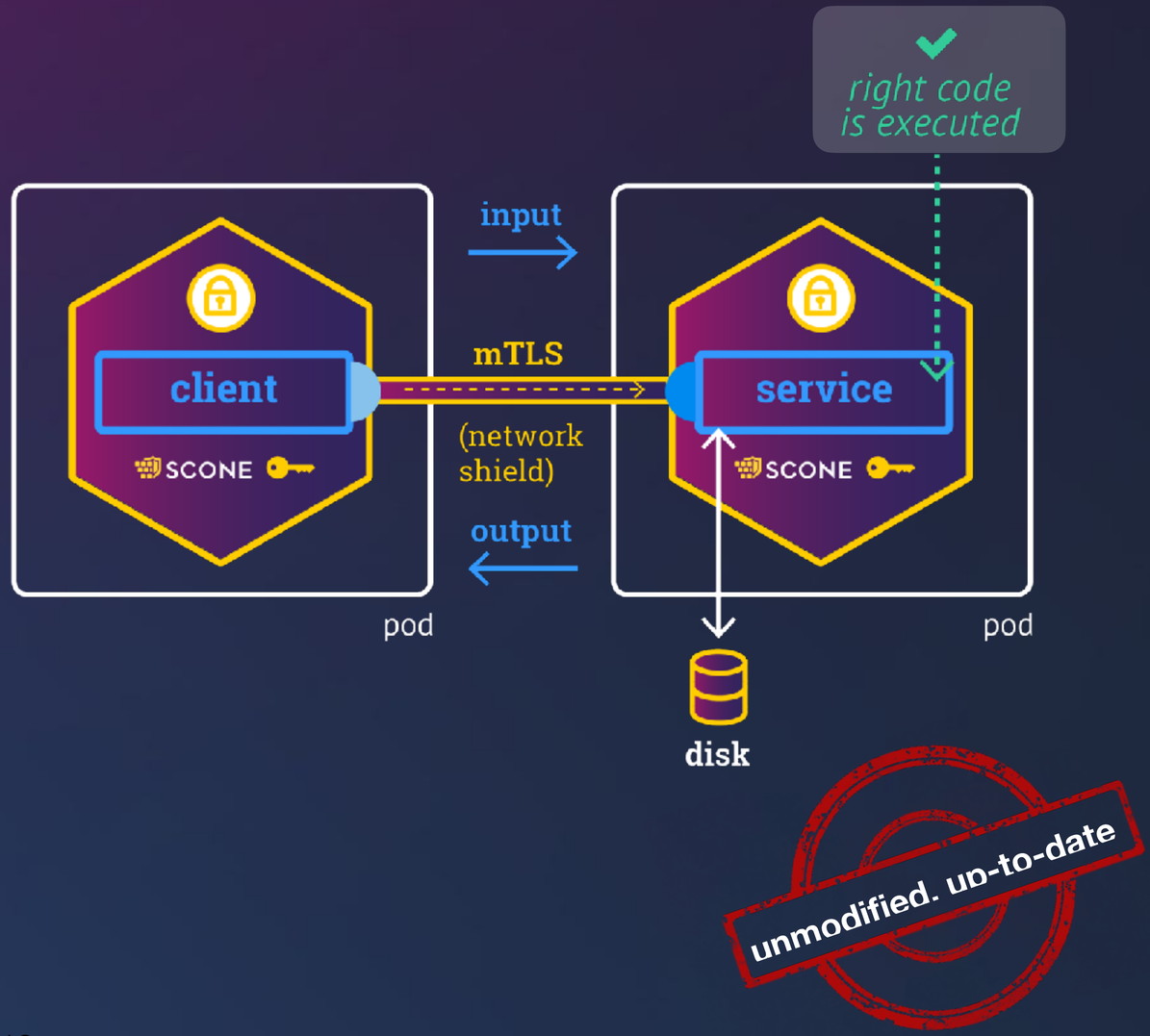
PROBLEM:

How to establish trust in
the CPU and the code in the enclave?

Attestation & Verification!

Confidential Execution

- Attestation & Verification
  - is the **right code** executed?
  - executed in the **right environment**?
  - **state at rest** is correct?
  - no known vulnerabilities in the **TCB**
    - TCB = Trusted Computing Base
  - the client is correct?
  - …



Enclave construction

- Someone may tamper with the enclave by modifying the code

```
int process_request(char *in, char *out)
{
  copy_msg(in, input_buf);
  if(verify_MAC(input_buf))
  {
    decrypt_msg(input_buf);
    process_msg(input_buf, output_buf);
    encrypt_msg(output_buf);
    copy_msg(output_buf, out);
    EEXIT(0);
  } else
    EEXIT(-1);
}
```

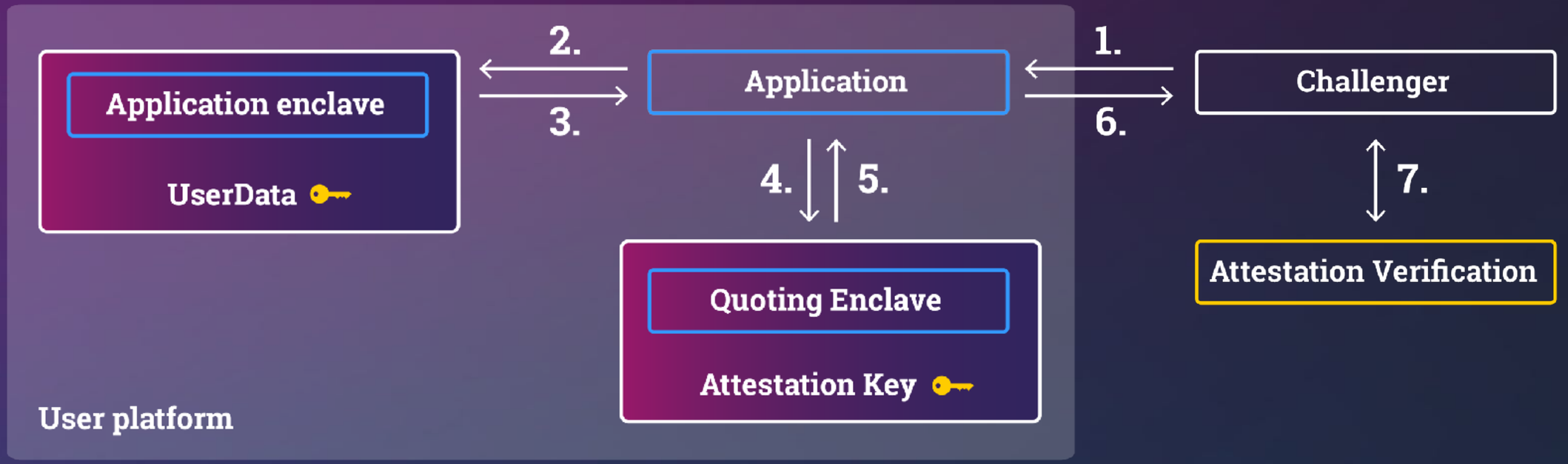
```
int process_request(char *in, char *out)
{
  copy_msg(in, input_buf);
  if(verify_MAC(input_buf))
  {
    decrypt_msg(input_buf);
    process_msg(input_buf, output_buf);
    copy_msg(output_buf, external_buf);
    encrypt_msg(output_buf);
    copy_msg(output_buf, out);
    EEXIT(0);
  } else
    EEXIT(-1);
}
```

Write unencrypted response to outside memory

Attestation & Verification

- We need to learn
  - what **code** is running? what is the initial state of an enclave?
  - what CPU / platform executes the program?
  - what is the disk state?
  - what are the **security versions** of the components of the TCB?
    - TCB = BIOS, CPU, firmware, „architectural enclaves“, …
  - how to communicate securely with the enclave (e.g., via TLS)?
  - …

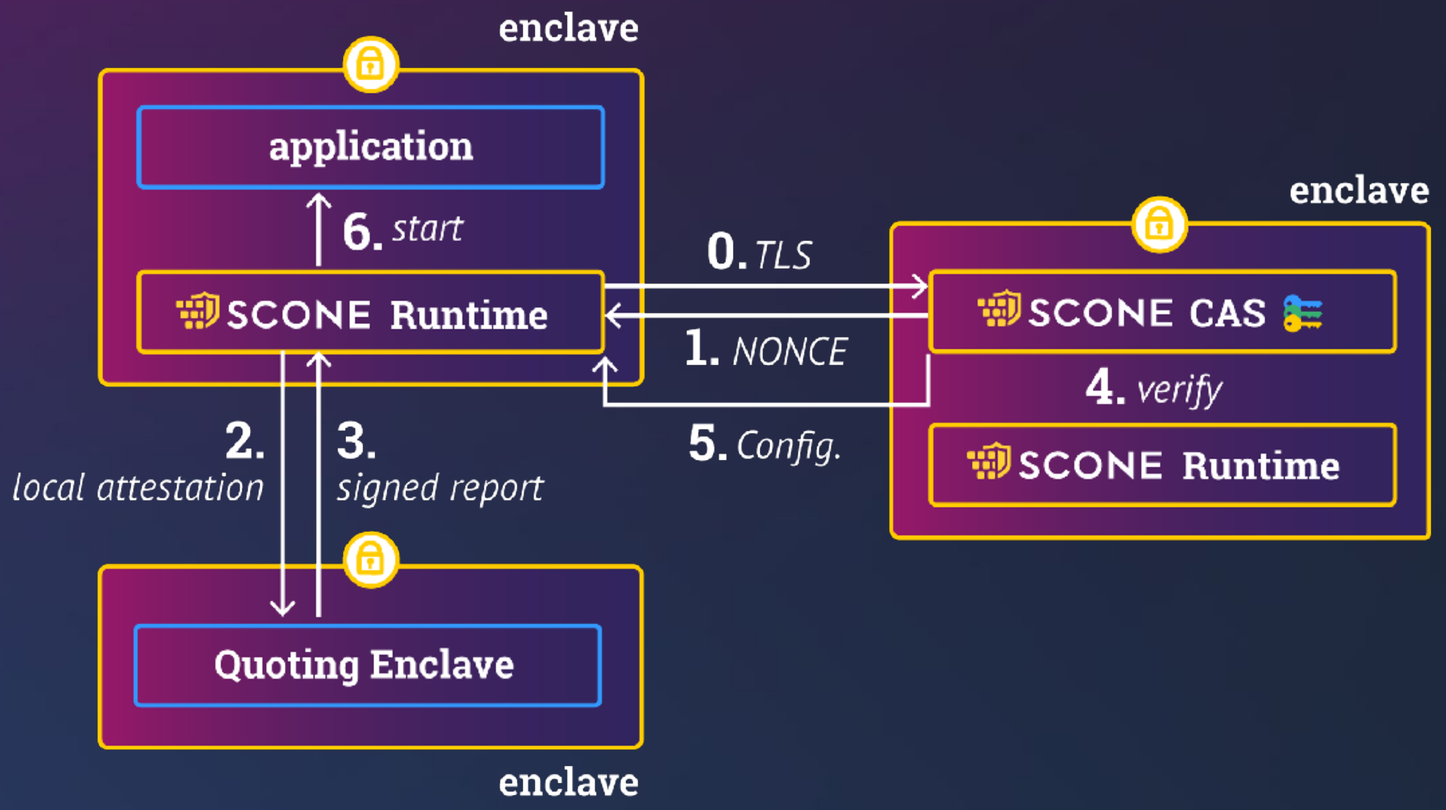
Intel SGX Attestation



- Attestation of enclave in SGX:
  - report generated by QE (an architectural enclave)
- Communication with challenger via application
- Application needs to be refactored!

SCONE Attestation

- **SCONE**:
  - no need to change application
- **Attestation flow:**
  - transparently performed by SCONE runtime
  - application gets configuration
    - arguments
    - environment variables
    - configuration files



Questions:

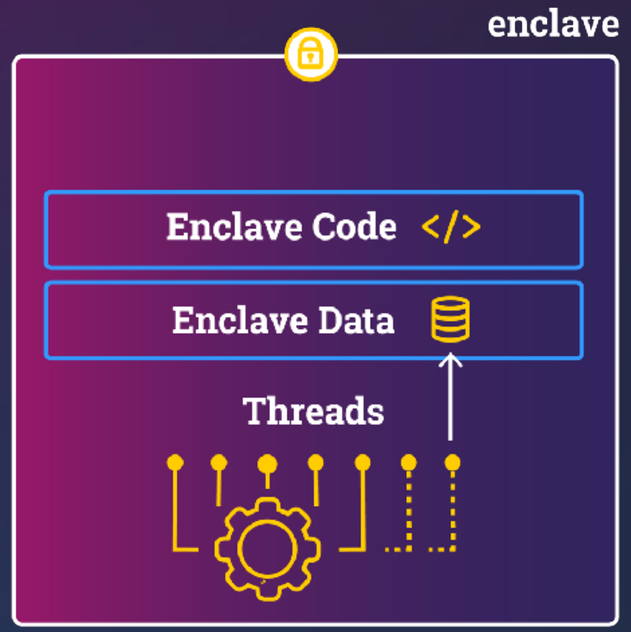
How can we trust an enclave?
If we do not trust the Hypervisor/OS/Hardware?
What is local / remote attestation?
How can we implement attestation?

Attestation Deep Dive

Attestation

Approach:

- ensure that the initial state of enclave is correct
  - create a signed report
- ensure that we run on real SGX hardware
  - hardware / firmware is up-to-date



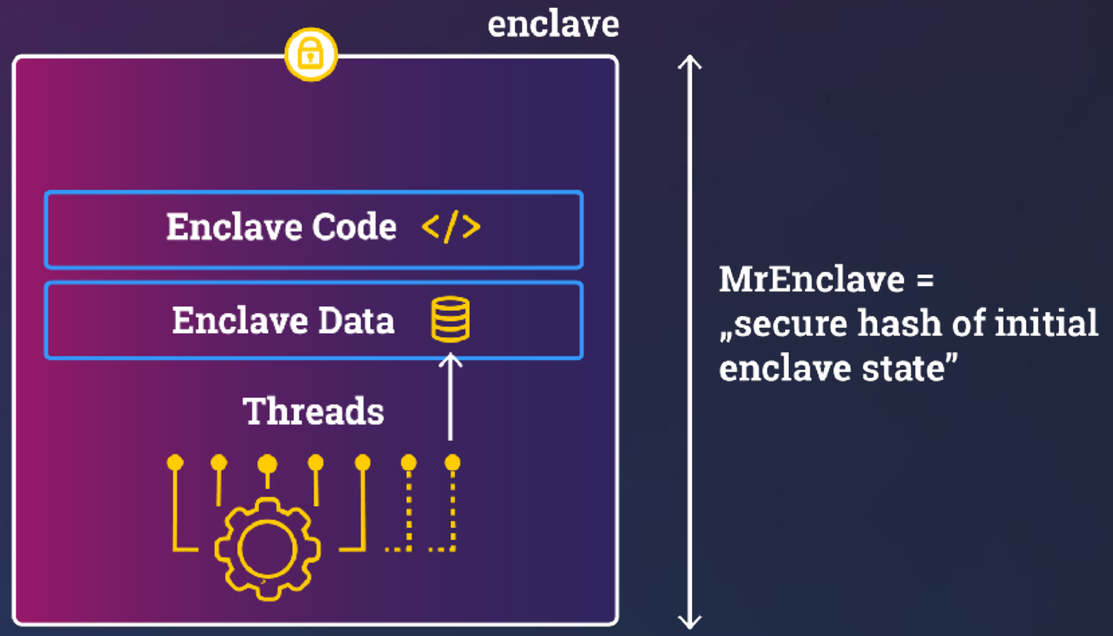
Enclave in correct initial state?

MrEnclave

Approach:

- ensure that the initial state of enclave is correct
  - create a signed report
- ensure that we run on real SGX hardware

- MRENCLAVE:
  - 256bit value
  - hashing page data and page metadata with SHA256

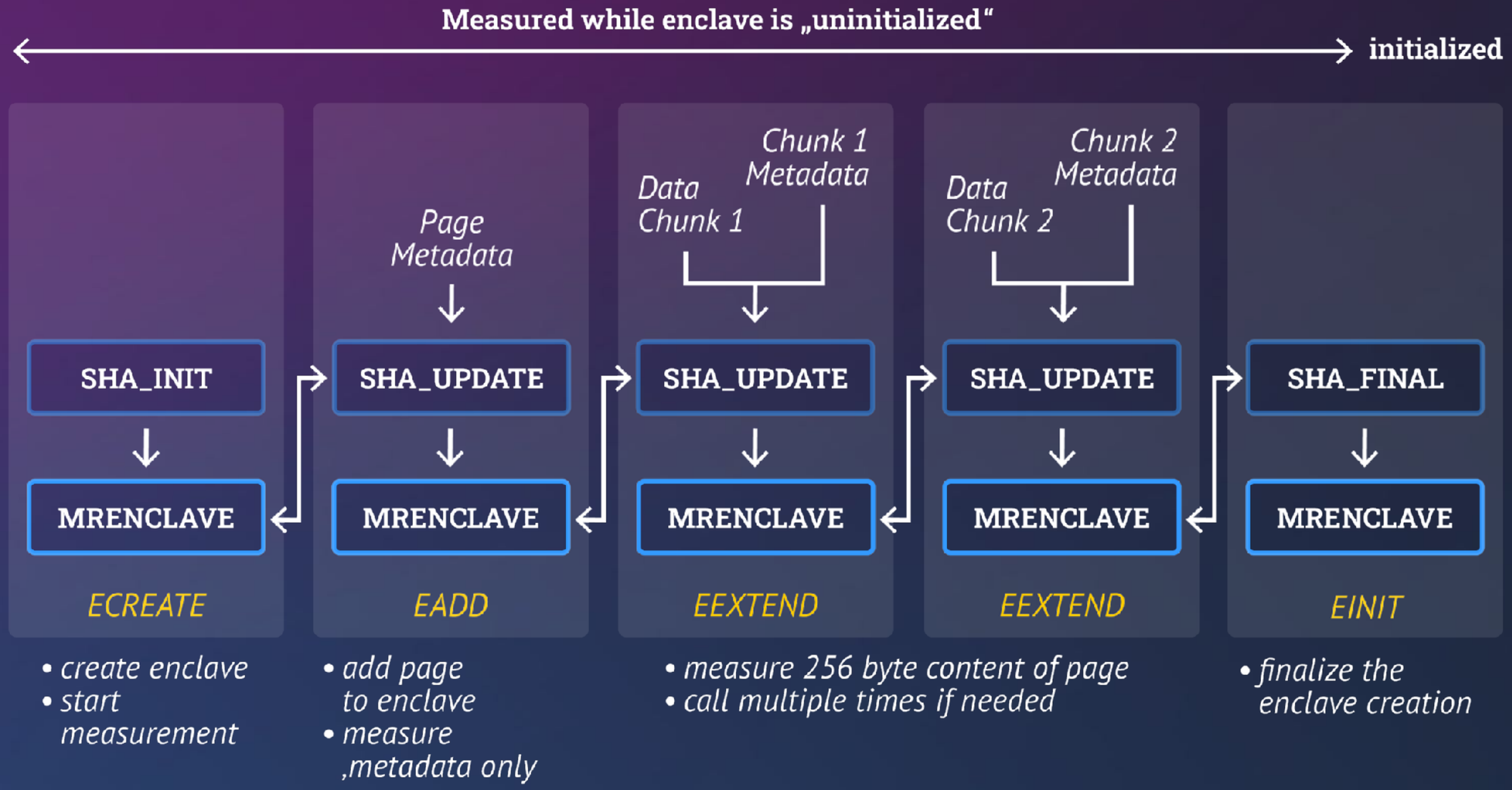


Enclave in correct initial state?

Enclave construction

- CPU calculates enclave's measurement hash during enclave construction
  - Each new page extends the hash with the page content and attributes (read/write/execute)
  - Hash computed with SHA256
  - Developer can decide not to measure all content
    - e.g., pages that are always set to 0, do not need to be measured
- Measurement is used to attest the enclave to a local or remote entity
  - …part of a signed report (aka quote)

MRENCLAVE: Measurement Flow



(Figure 35-2 in Intel Manual 325462-sdm-vol-1-2abcd-3abcd.pdf)

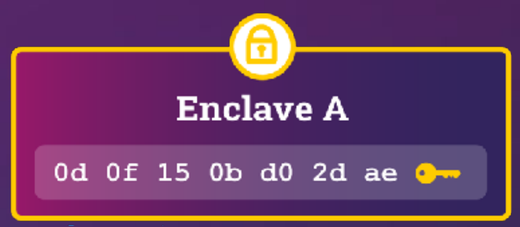
Enclave attestation

- Is my code running on remote machine intact?
- Is code really running inside an SGX enclave?
- Local attestation
  - Prove enclave's identity (=measurement) to another enclave on the same CPU / same server (SGX supports multi-socket servers)
- Remote attestation
  - enclave's identity to a remote party
- Once attested, an enclave can be trusted with secrets

Attestation: Learn unique ID of Enclave A

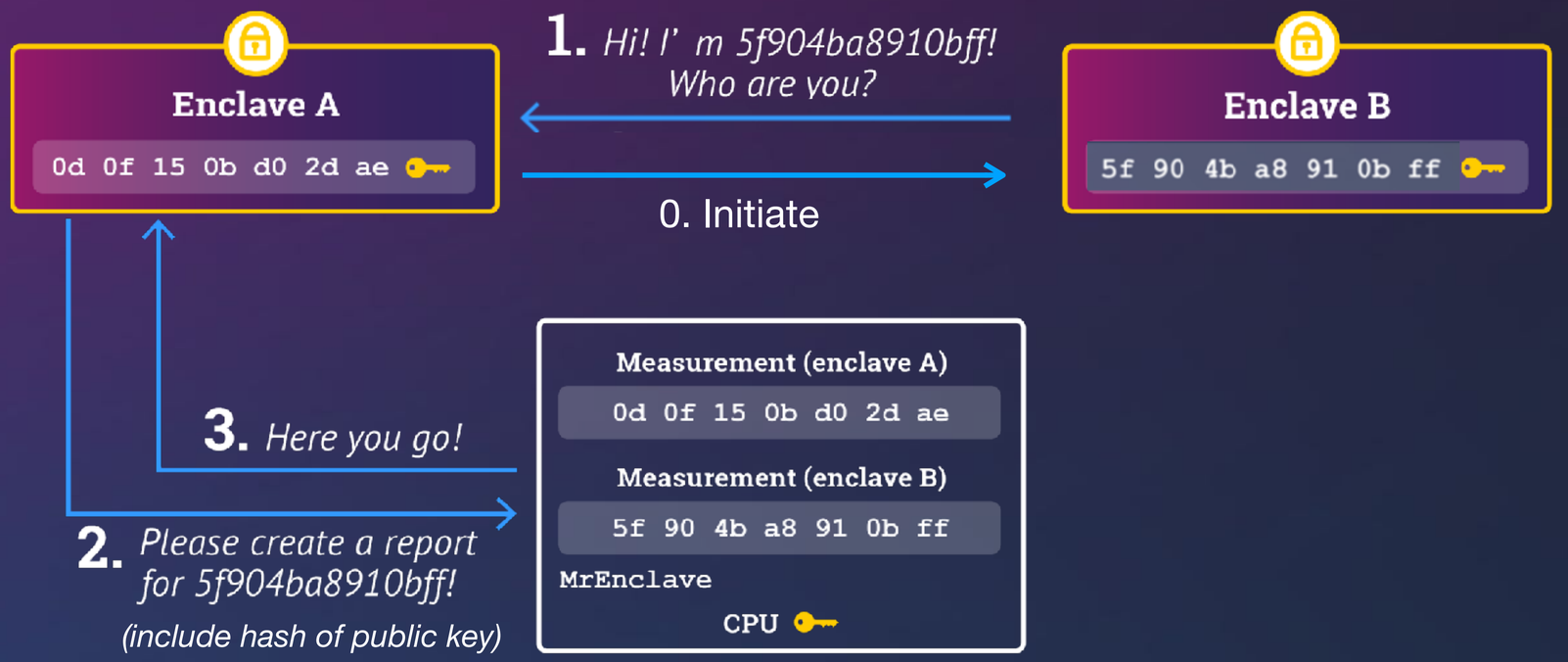
Prove identity of A to a local enclave B (= Quoting Enclave)

0. *Create public/ private keypair PublicKey - PrivateKey*



Local attestation I

Prove identity of A to a local enclave B (= Quoting Enclave)

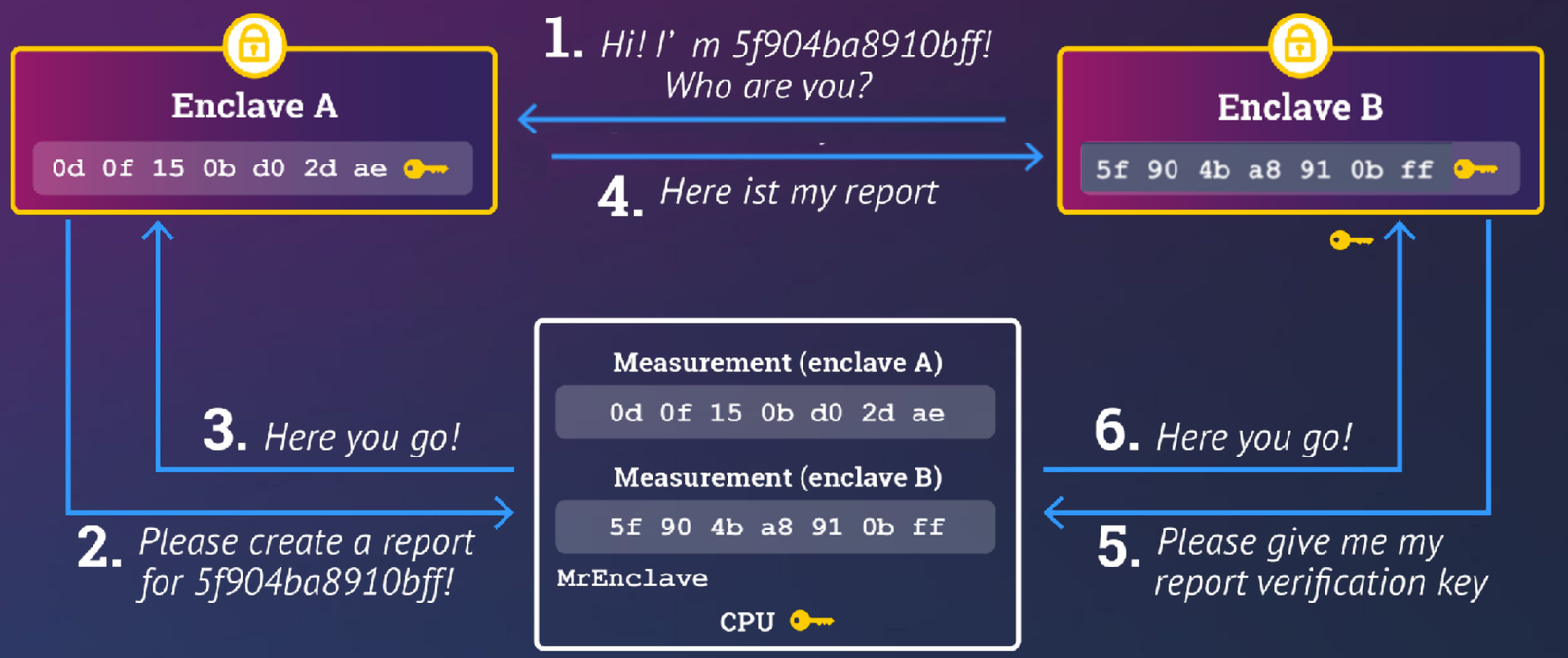


1. Target enclave B measurement is required for key generation
2. Report contains information about target enclave B, including its measurement
3. CPU fills in the report and creates a MAC using the report key, which depends on random CPU fuses and the target enclave B measurement

...

Local attestation II

- Prove identity of A to a local enclave B (= Quoting Enclave)



...

4. Report sent back to target enclave B
5. Verify report by CPU to check that it was generated on the same platform, i.e., its MAC was created with the same report key (available only on the same CPU)
6. Check the MAC received with the report and do not trust A upon mismatch

Remote Attestation

- Transform a local report to a remotely verifiable “quote”
  - asymmetric instead of symmetric cryptography
  - signs the report with an **attestation key**
- Quote generated by quoting enclave (QE)
  - often, an architectural enclaves provided by Intel
    - one can also define quoting enclaves independent of Intel
  - Executes locally on user platform