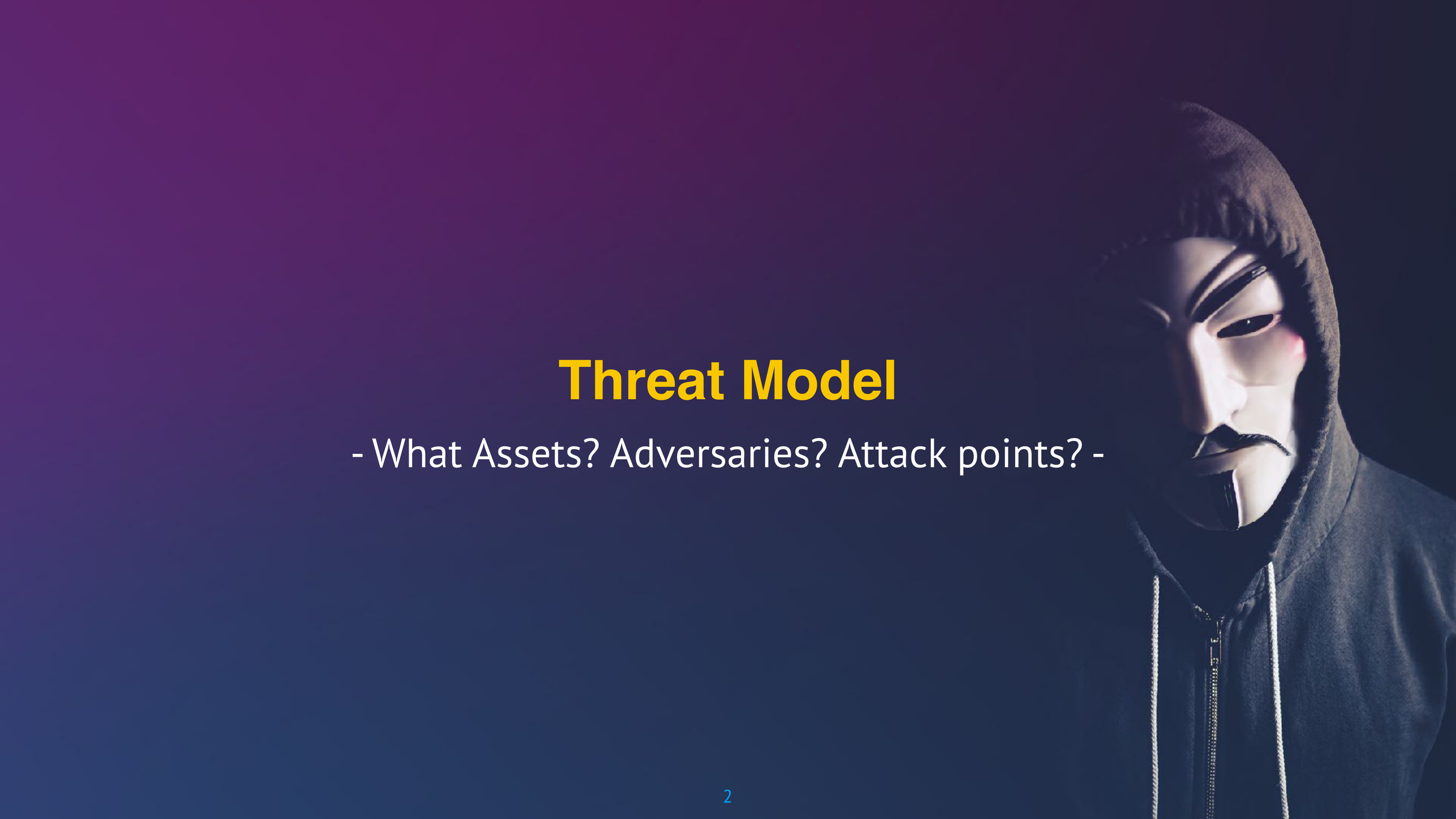
CONFIDENTIAL COMPUTING

Attestation

Prof. Dr. Christof Fetzer

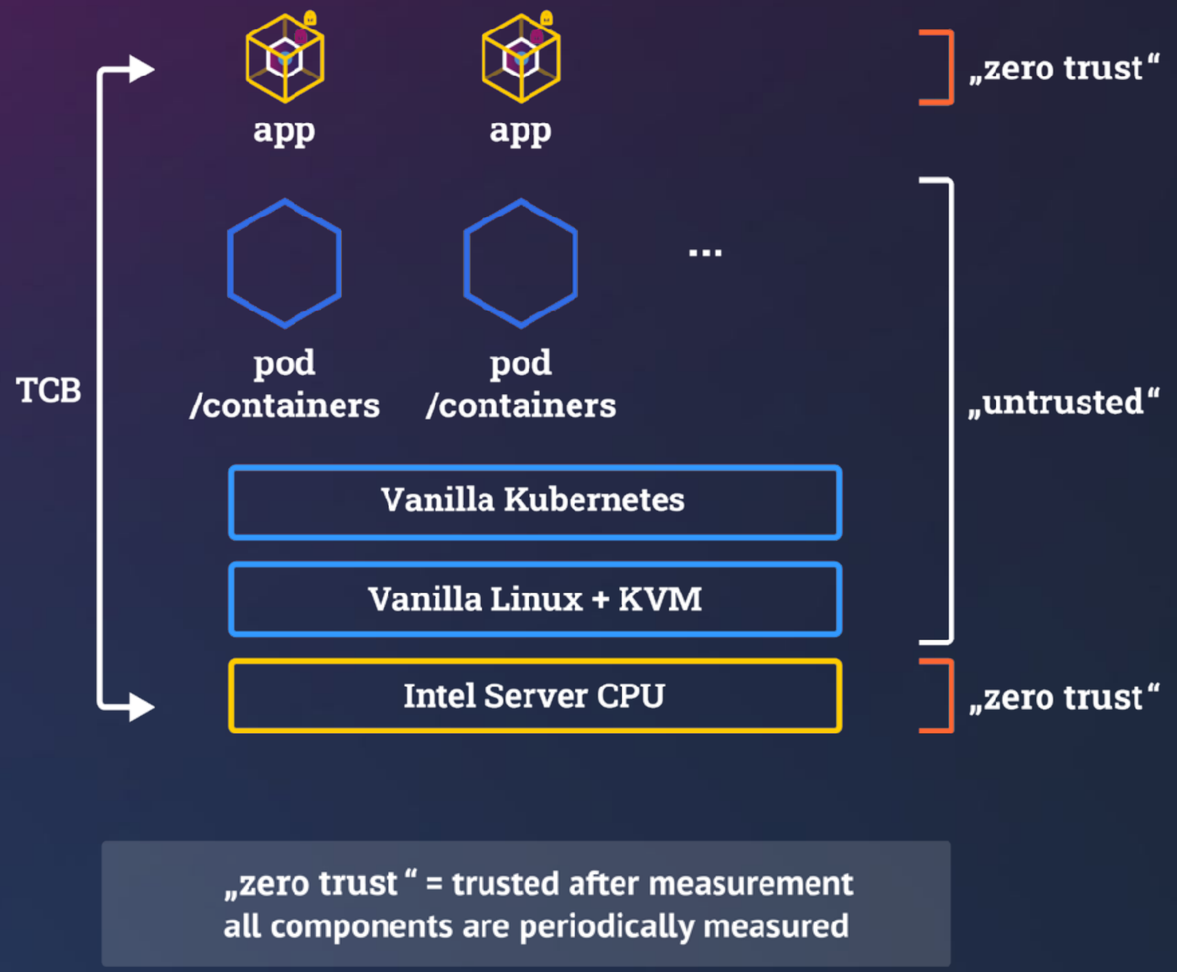
Threat Model

- What Assets? Adversaries? Attack points? -



What do we trust?

- CPU with TEE support
  - after attestation & verification
- Services of application
  - after attestation & verification



Threat Model

A1) **Unprivileged Software Adversary**

A2) **System Software Adversary**

A3) **Startup Code/SMM Software Adversary**

A4) **Network Adversary**

A5) **Software Side-Channel/Covert-Channel Adversary**

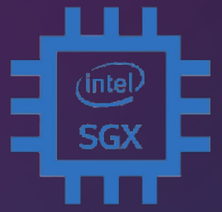
A6) **Simple Hardware Adversary**

A7) ***Roll-Back State Adversary***

A8) ***CVE Adversary***

A9) ***Insider attacks from Security Team***

https://www.intel.com/content/www/us/en/security/security-practices/secure-development-practices/threat-modeling.html

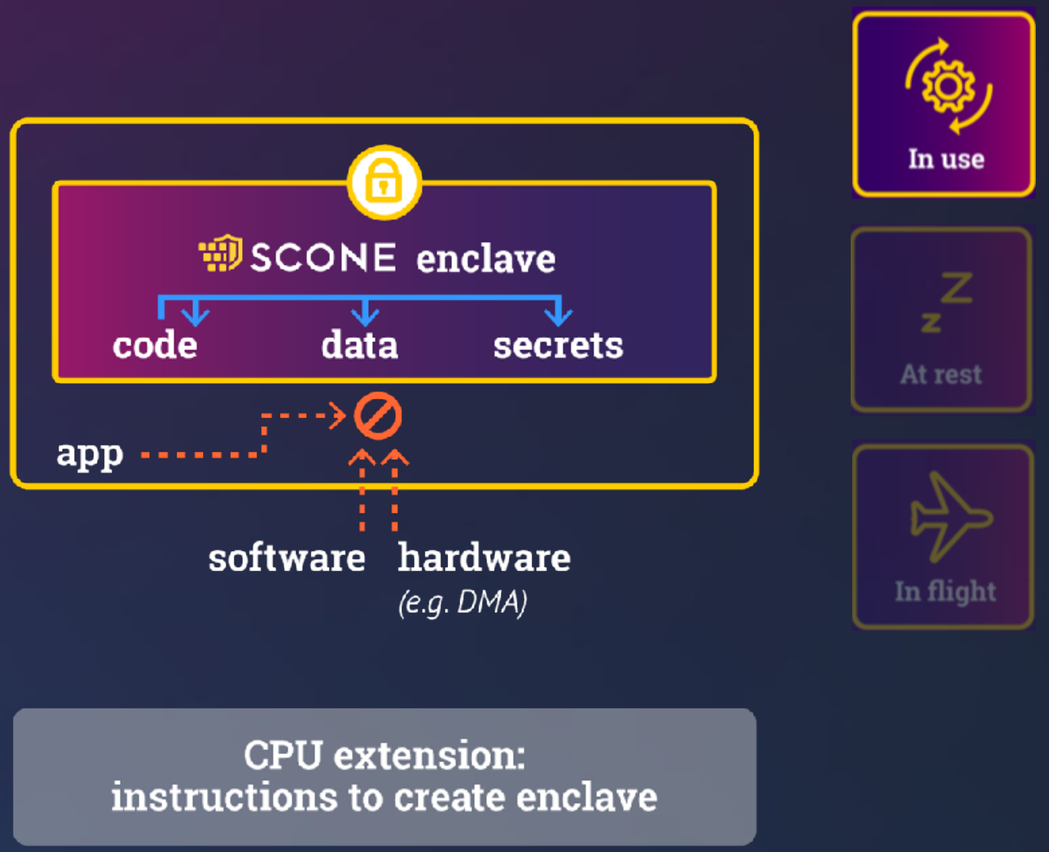


SCONE (SECURE CONTAINER ENVIRONMENT)

- SCONE Attestation with SGX -

Enclaves

- **Protect data/code/secrets in use**
  (i.e, in main memory):
  - run application code in encrypted memory region (aka **enclave**)
  - only code in enclave can access region

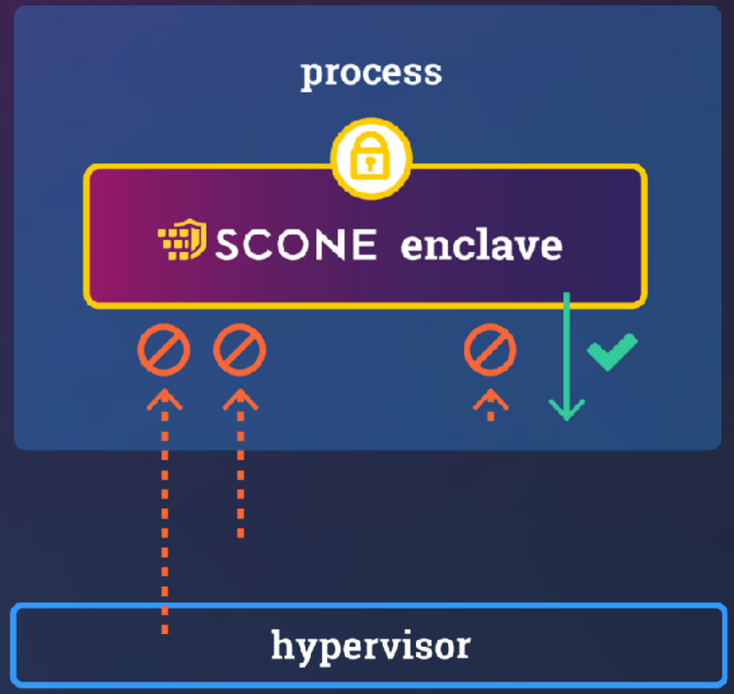


SCONE: Secure CONtainer Environment
- platform for Confidential Computing

https://sconedocs.github.com

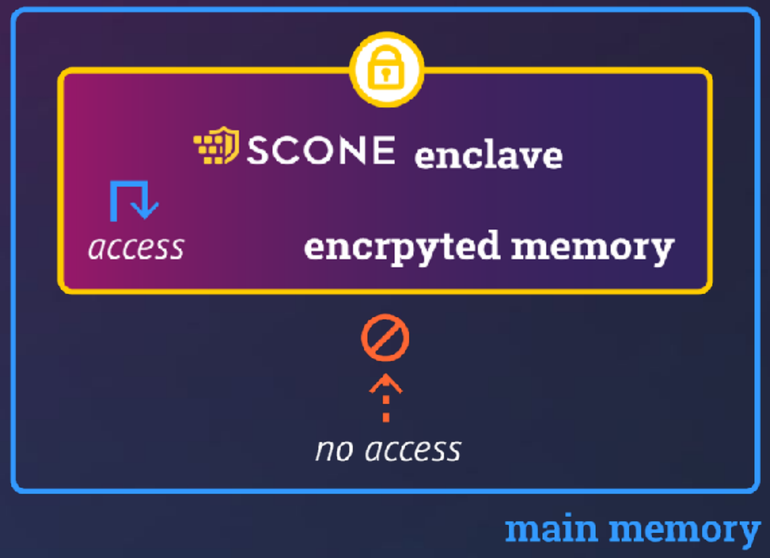
Enclave memory

- 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



Trusted Execution Environment (TEE)

- Enclaves are a TEE:
  - CPU instruction set extension
  - service runs inside an encrypted memory region, aka, **enclave**
    - **random key** — generated by CPU, or
    - **external key** — given by external entity - **can we trust this key**?
  - only code running inside of enclave can see content
- **Hardware with different granularities:**
  - **enclave**: process runs inside encrypted memory (Intel SGX)
  - **encrypted VM**: a whole VM (AMD SEV, ARM Realms, Intel TDX)
    - can be used to implement enclaves



PROBLEM:

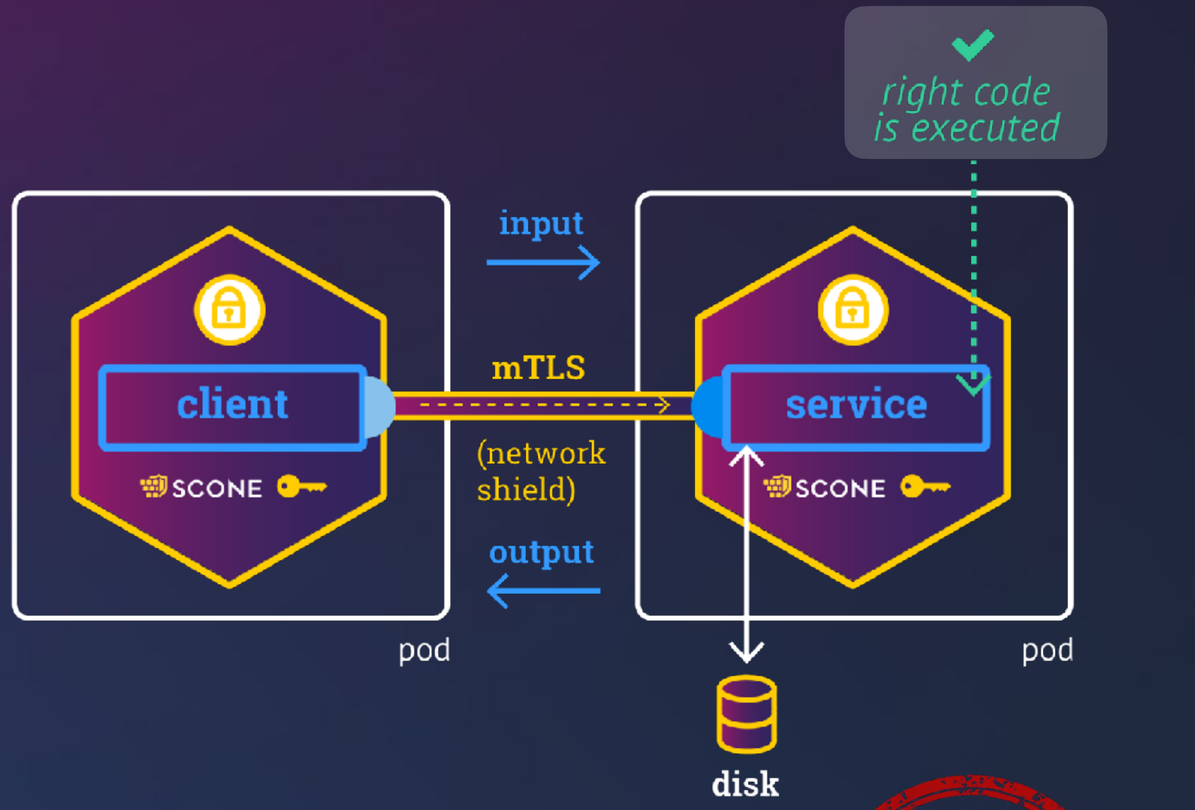
How to establish trust in
the CPU and the code in the enclave?

Attestation & Verification!

- „FaceID for applications“ -

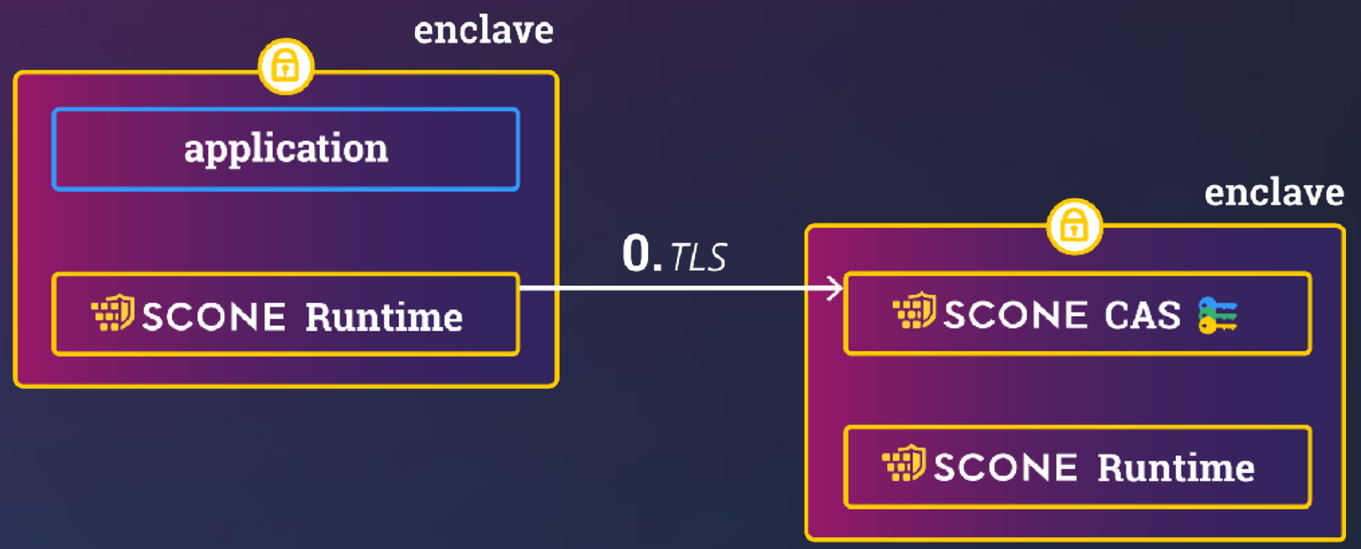
Confidential Execution

- 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?
  - …



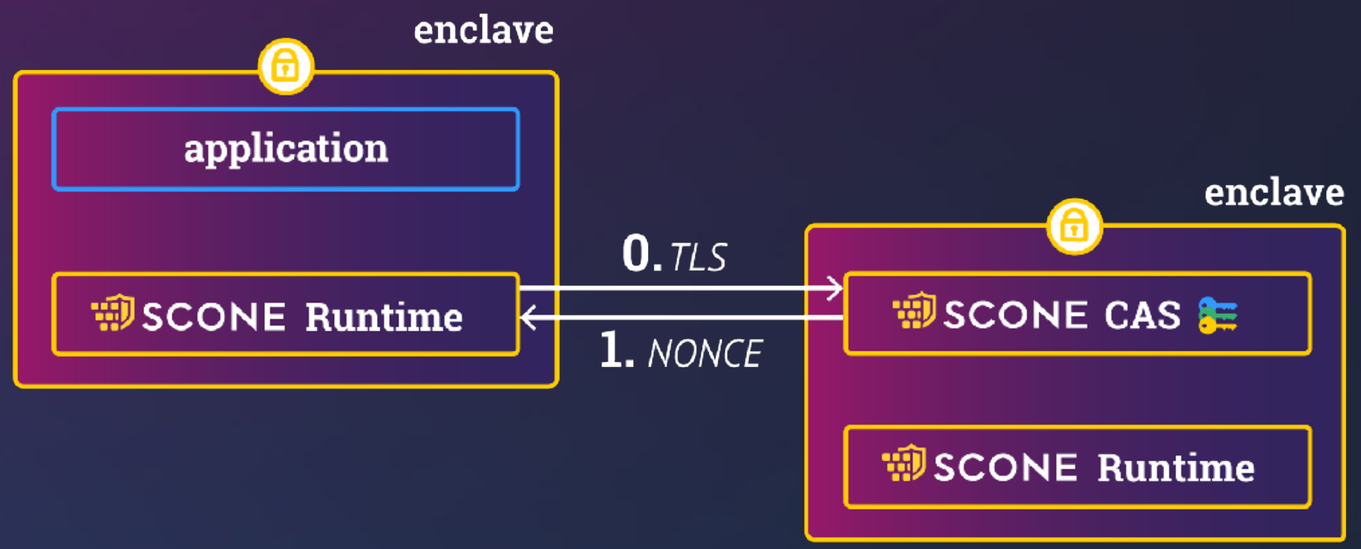
SCONE Attestation

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



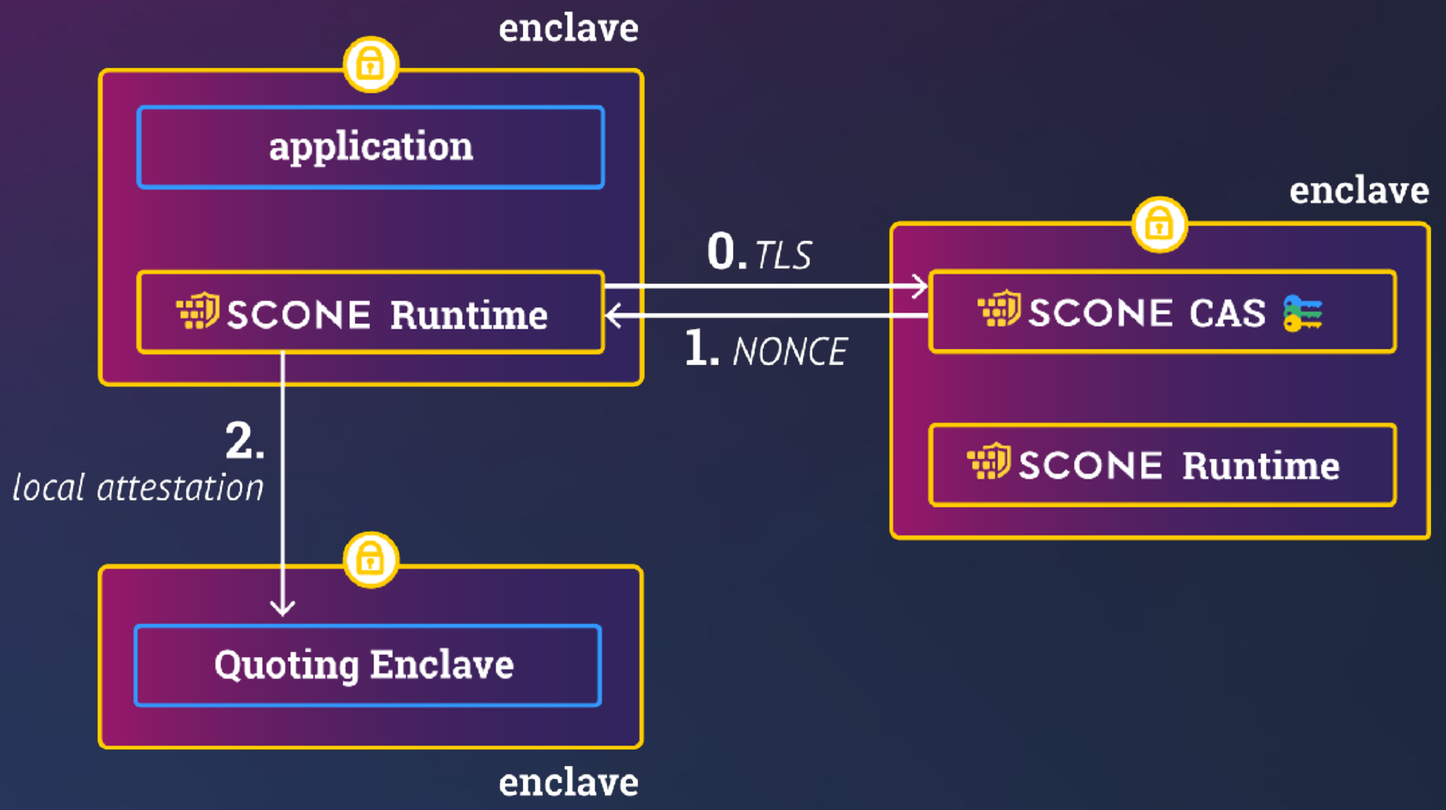
SCONE Attestation

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



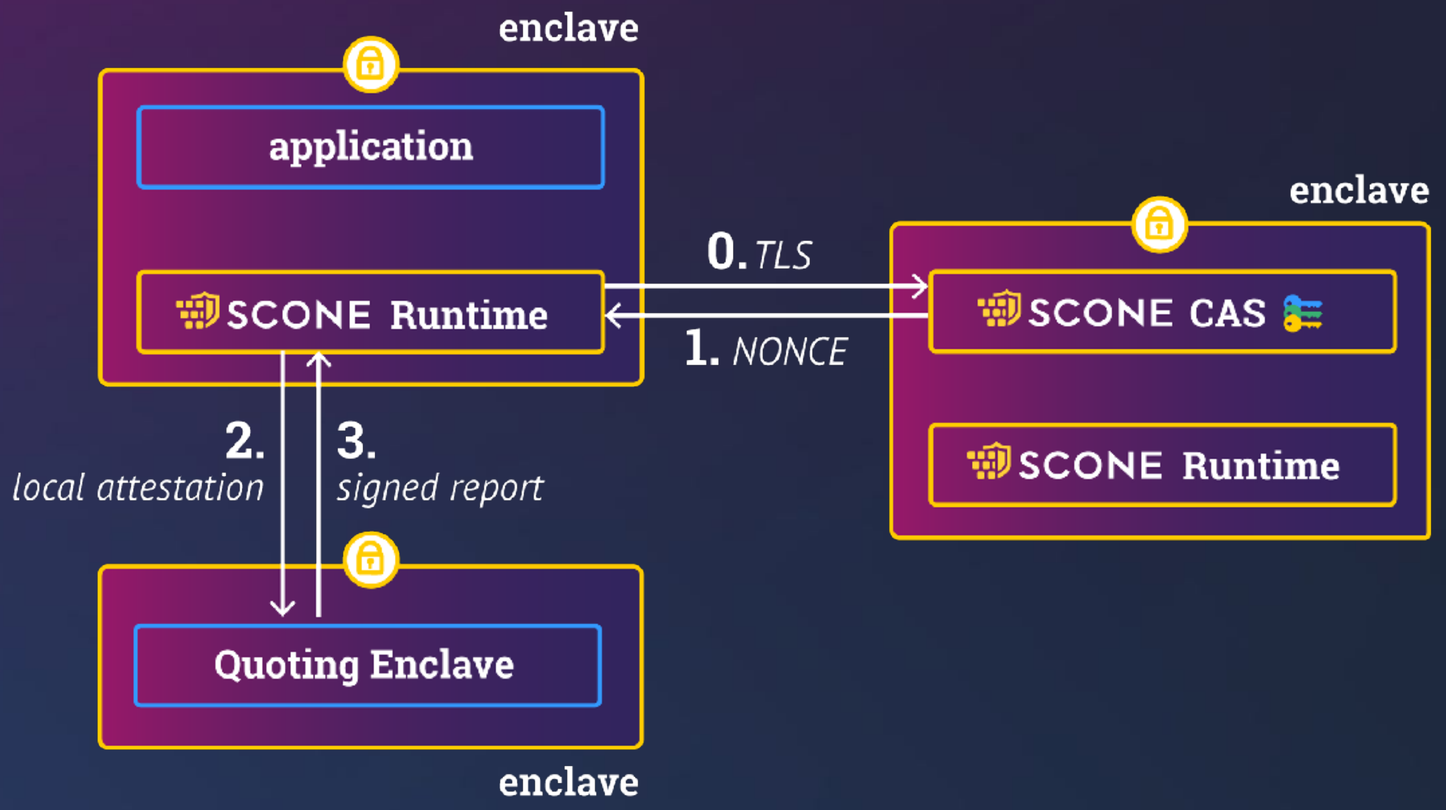
SCONE Attestation

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



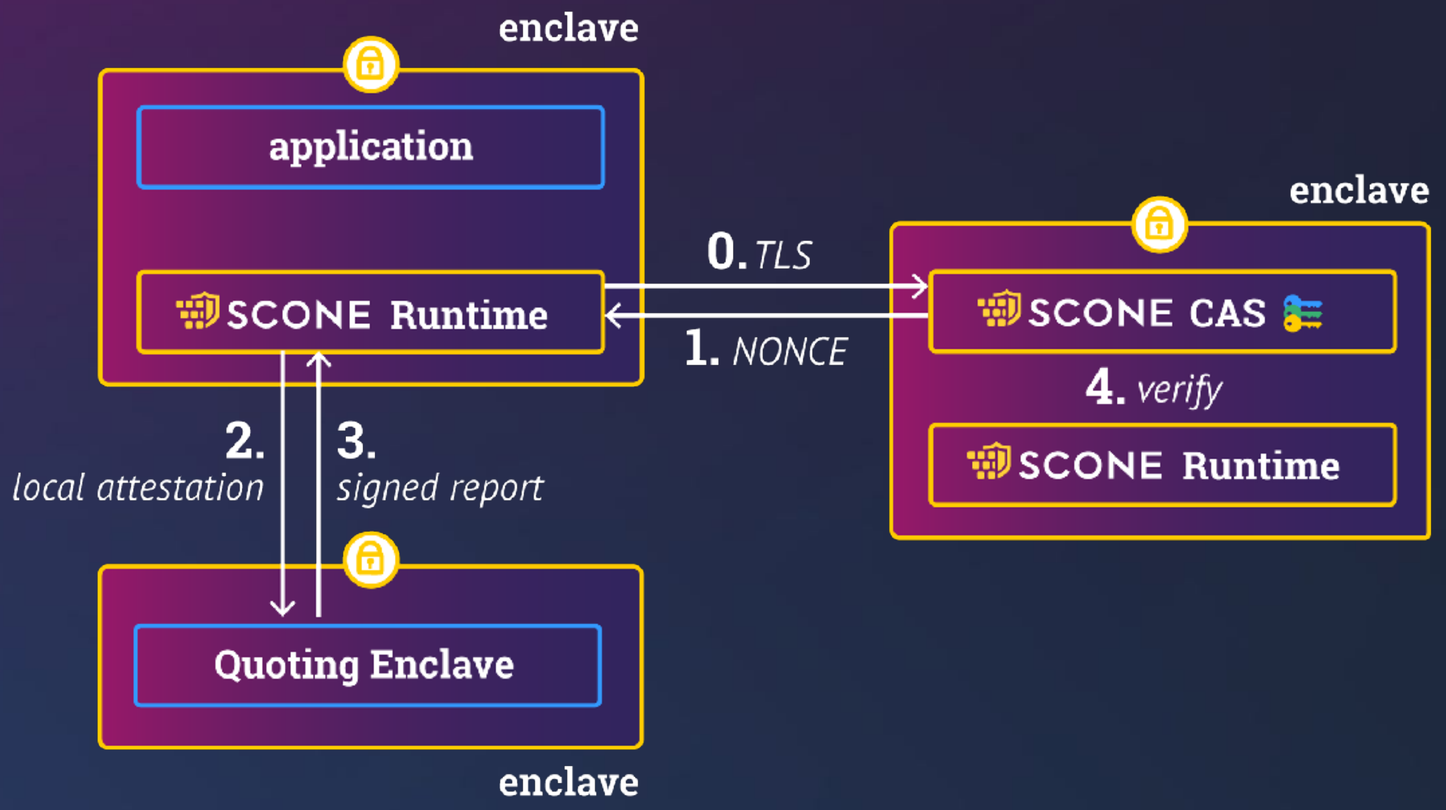
SCONE Attestation

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



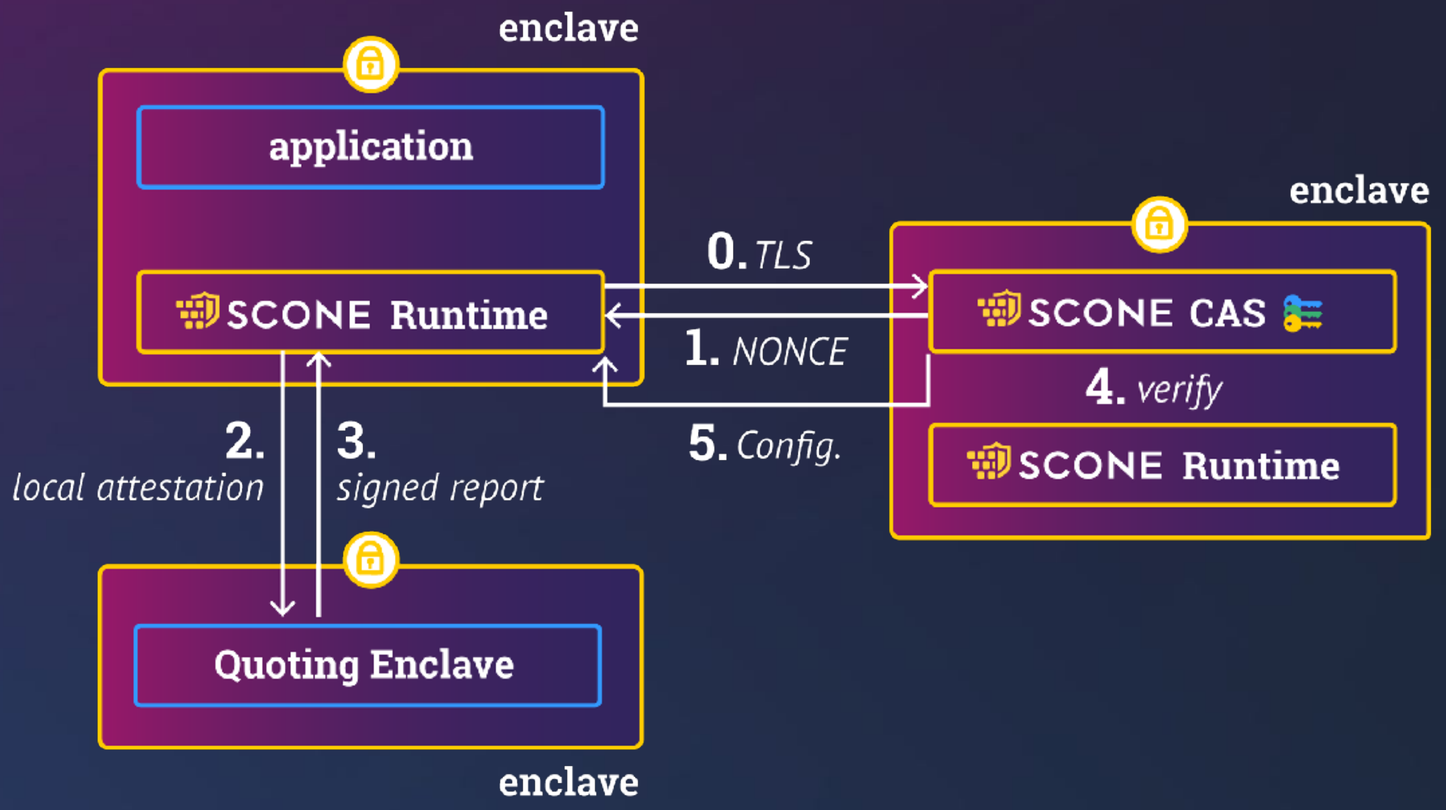
SCONE Attestation

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



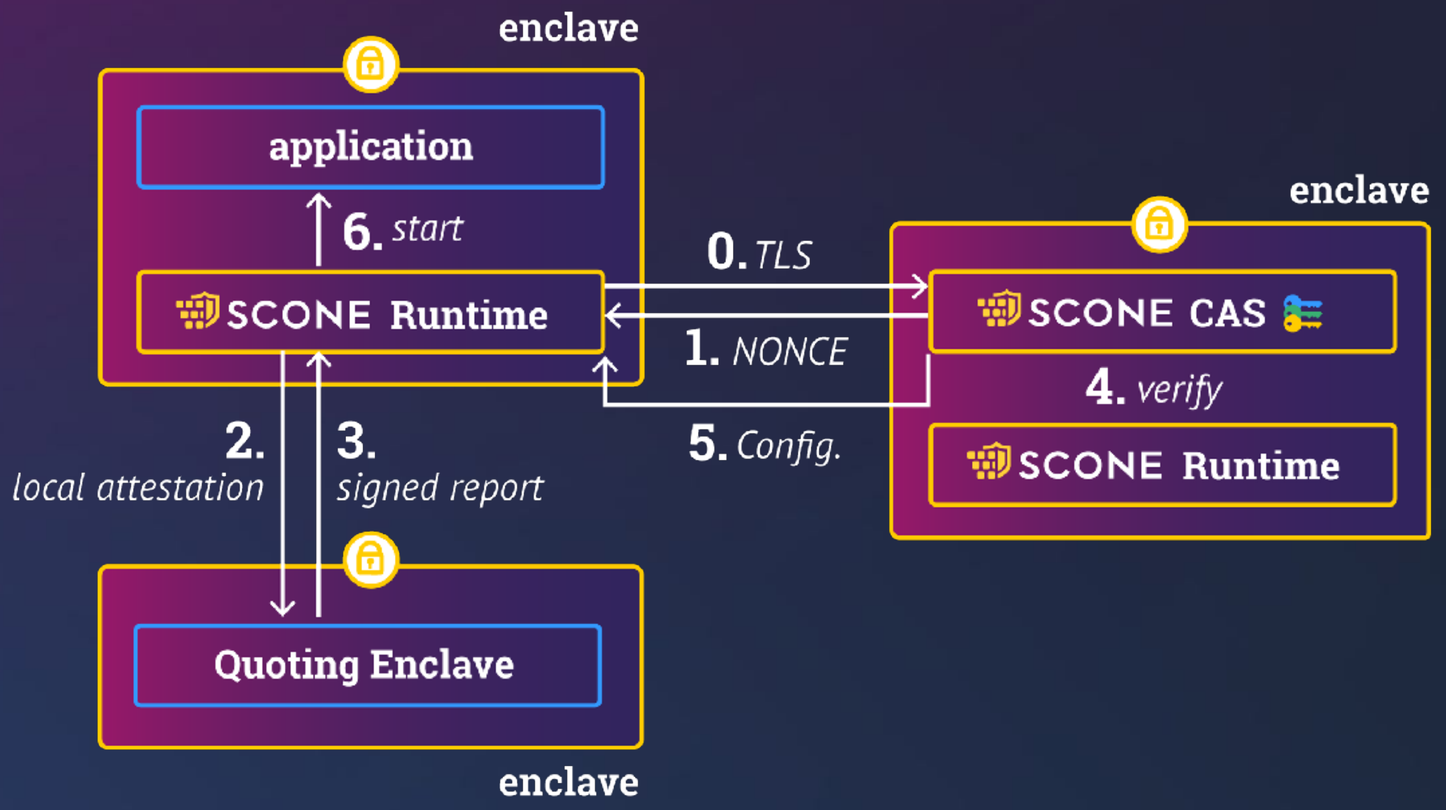
SCONE Attestation

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

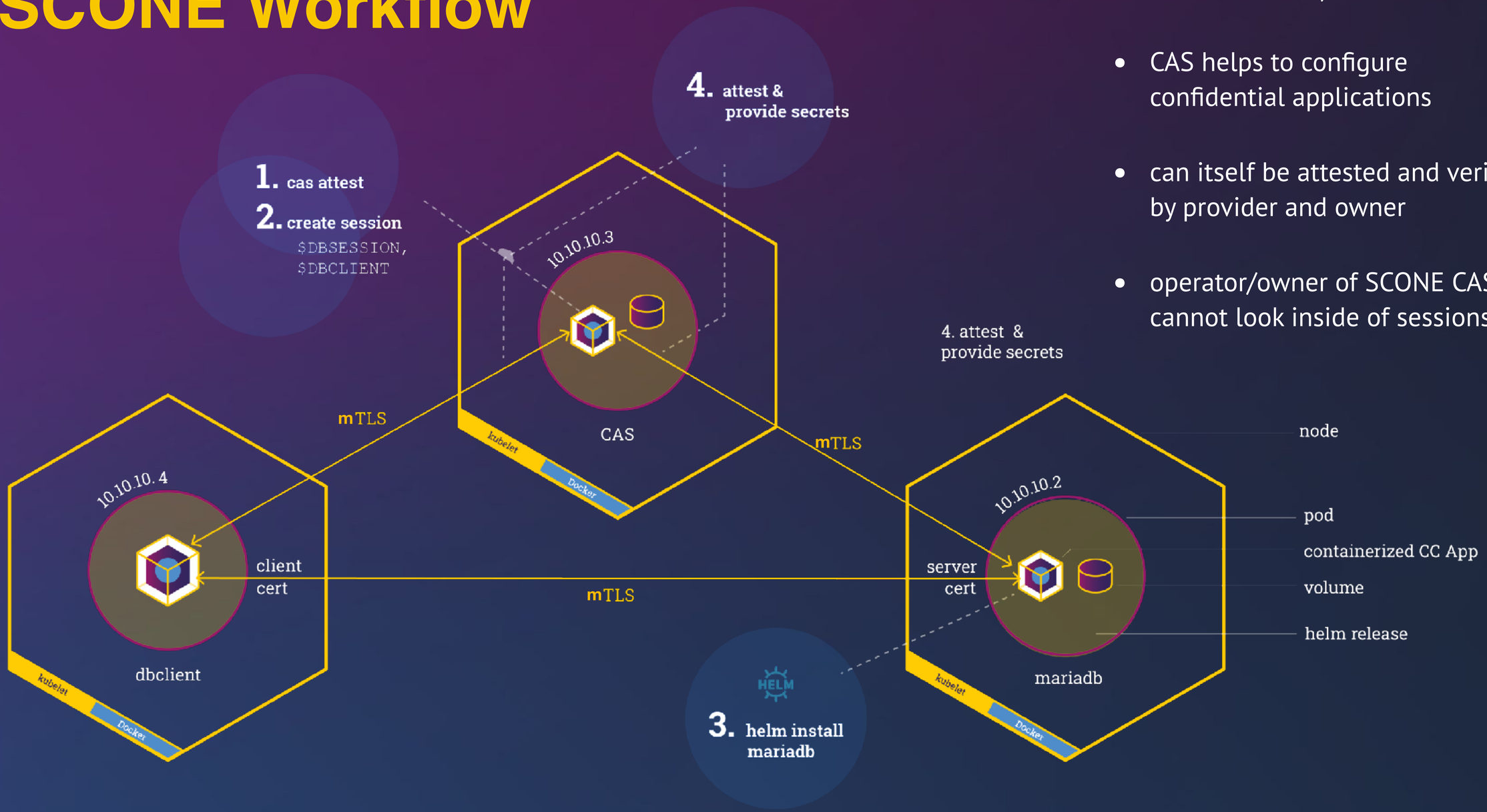


SCONE Attestation

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



SCONE Workflow



SCONE CAS (Configuration and Attestation Service):

- CAS helps to configure confidential applications
- can itself be attested and verified by provider and owner
- operator/owner of SCONE CAS cannot look inside of sessions