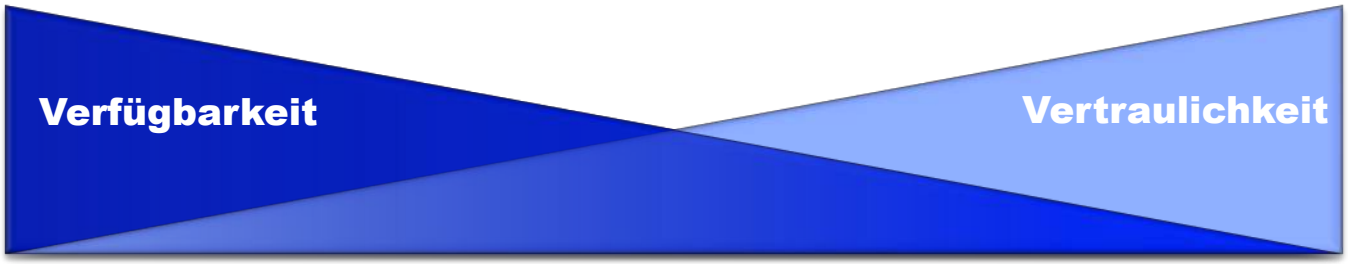
Hardware-basierte Sicherheit: Trusted Computing, TPM & Intel SGX

Problemstellung

- Ziel:
  - Auslagerung von Teilen der IT-Infrastruktur in die Cloud
- Sicherheits-Problem:
  - Datenspeicherung / -verarbeitung außerhalb der eigenen physischen Kontrolle
- Lösungsansätze:
  - organisatorische:
    - vertragliche Regelungen
    - Audits
  - technische!

- Ziel:
  - Auslagerung der **Datenspeicherung**
    - Backup
    - Archiv
  - Verarbeitung weiter hausintern

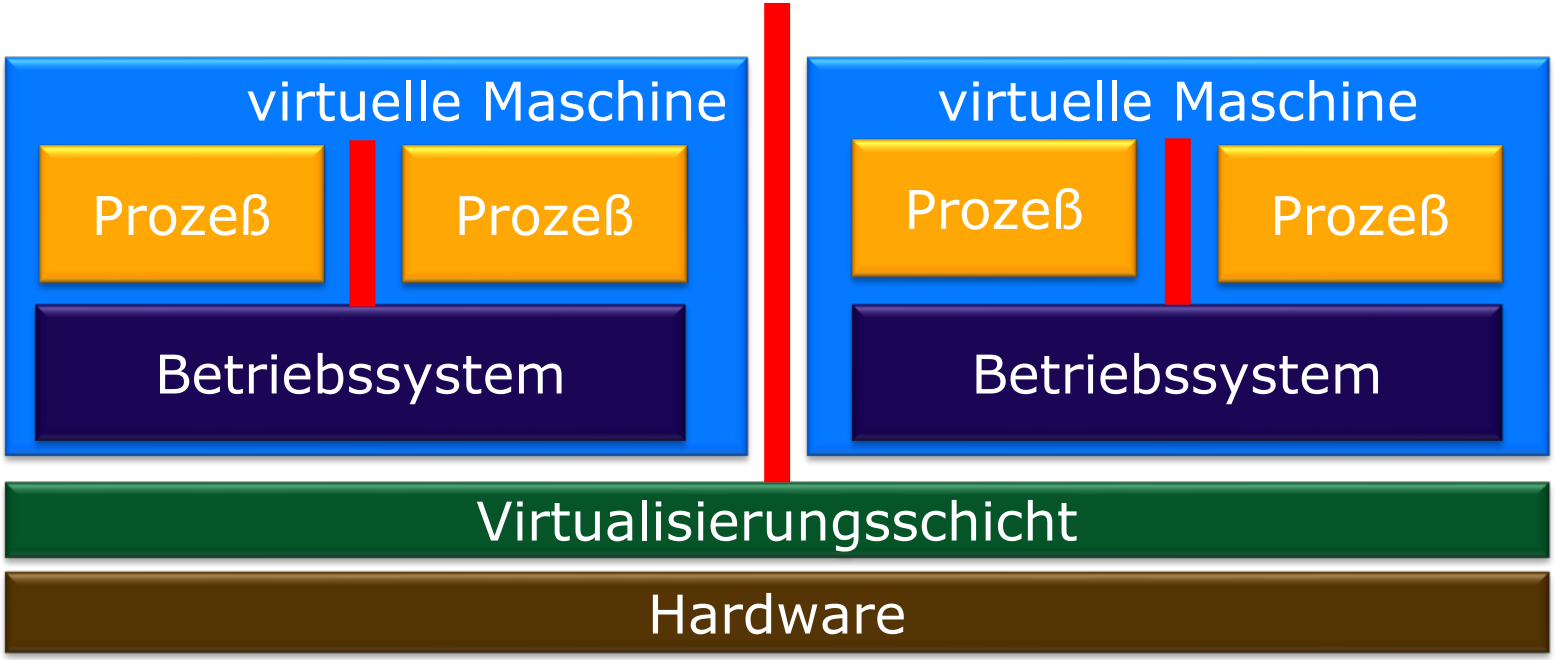
- Lösung:
  - Verschlüsselung der Daten ***vor*** Übermittlung in die Cloud
  - ggf. Aufteilung auf mehrere Cloud-Anbieter
    - *k*-aus-*n* Verfahren:
      - Schreiben: Aufteilung auf *n* Cloud-Anbieter
      - Lesen: *k*-Teile notwendig



$k=1$ $k=n$

- Ziel:
  - Auslagerung der Daten**verarbeitung**
    - ressourcenintensive Berechnungen
    - Big-Data
    - kostengünstige Skalierbarkeit
    - …

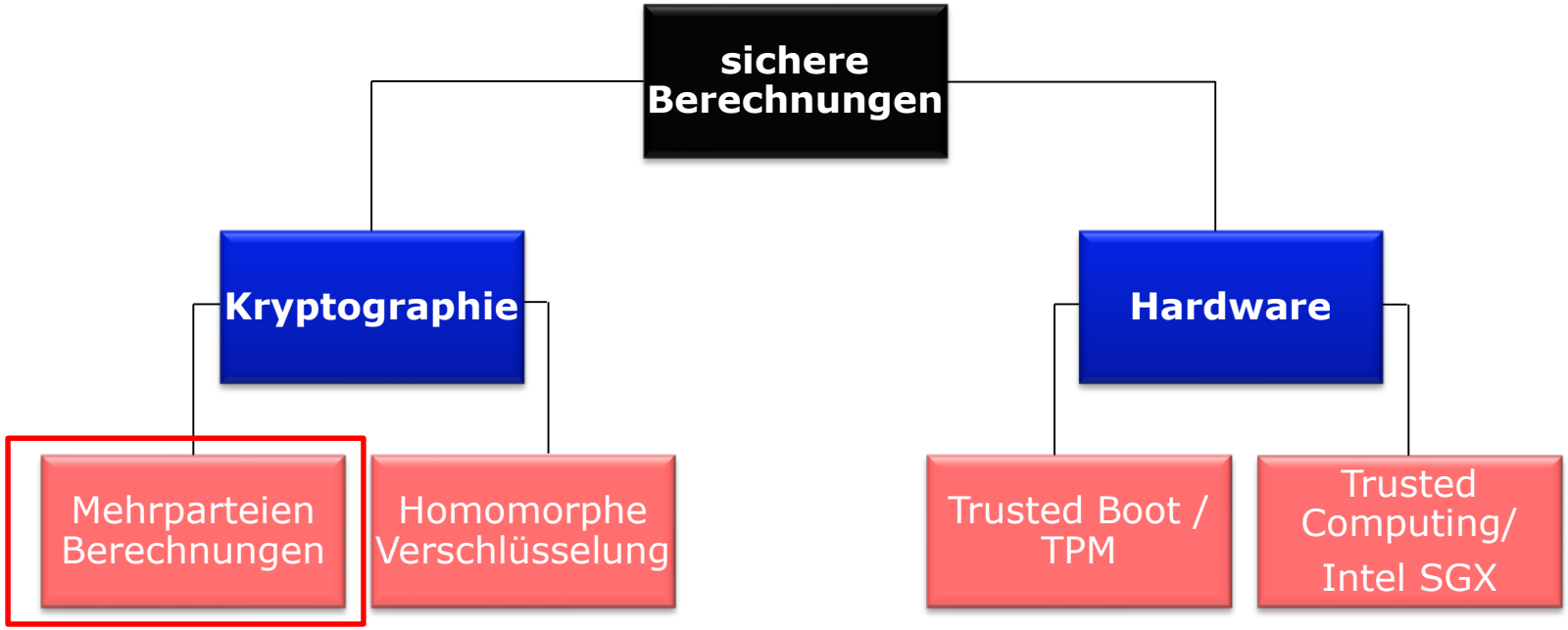
Infrastructure as a Service— Status quo



- Separierung durch Software
- große Trusted Computing Base (TCB):
  - Hardware, Virtualisierung (Hypervisor), Betriebssystem, Anwendungssoftware
- **kein** Schutz gegenüber Betreiber der Hardware

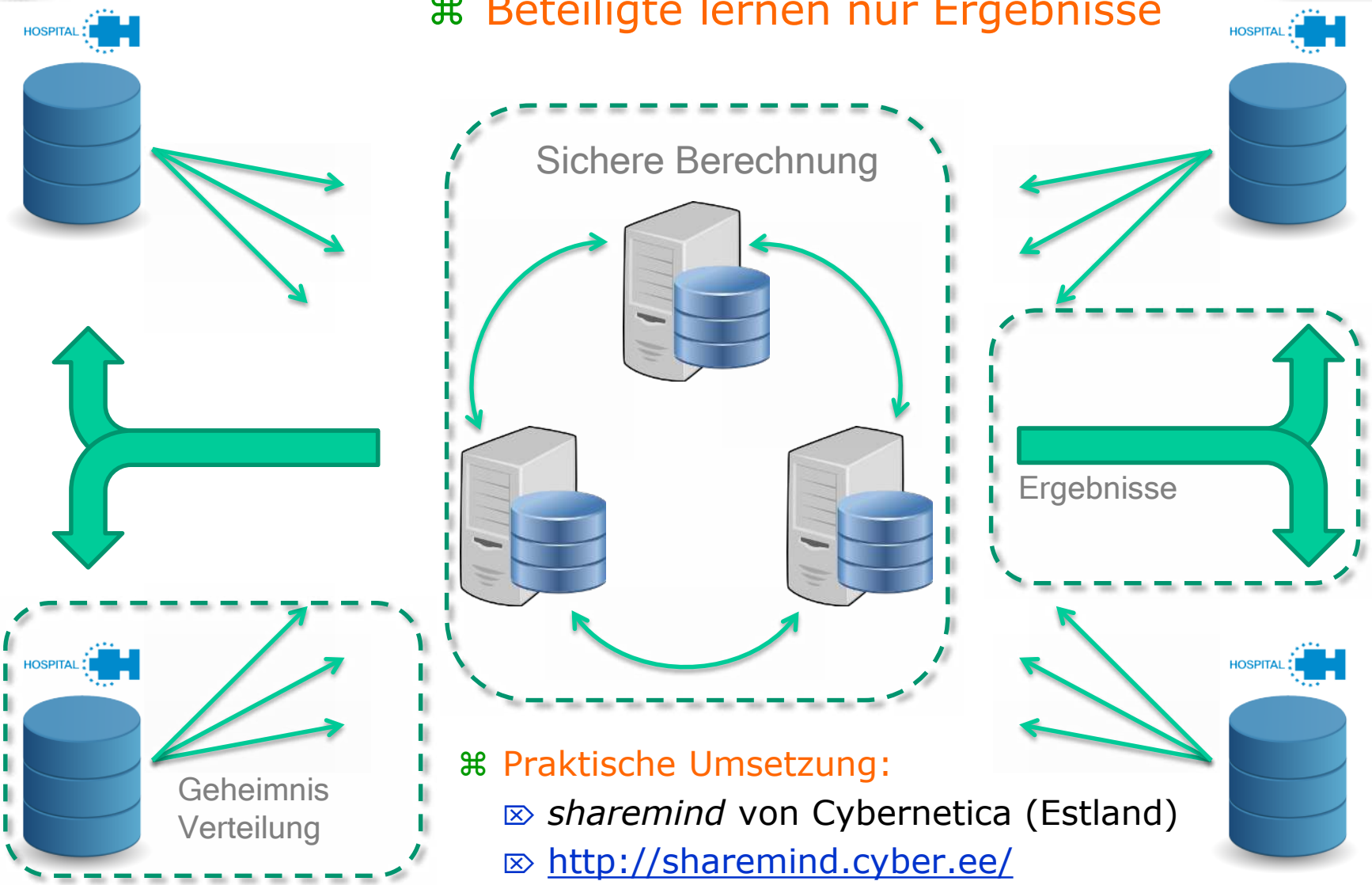
- Ziel:
  - Auslagerung der Datenverarbeitung
    - **Schutz auch gegenüber dem Betreiber!**

- Lösungsansätze:



Sichere Mehrparteien-Berechnungen

⌘ Beteiligte lernen nur Ergebnisse

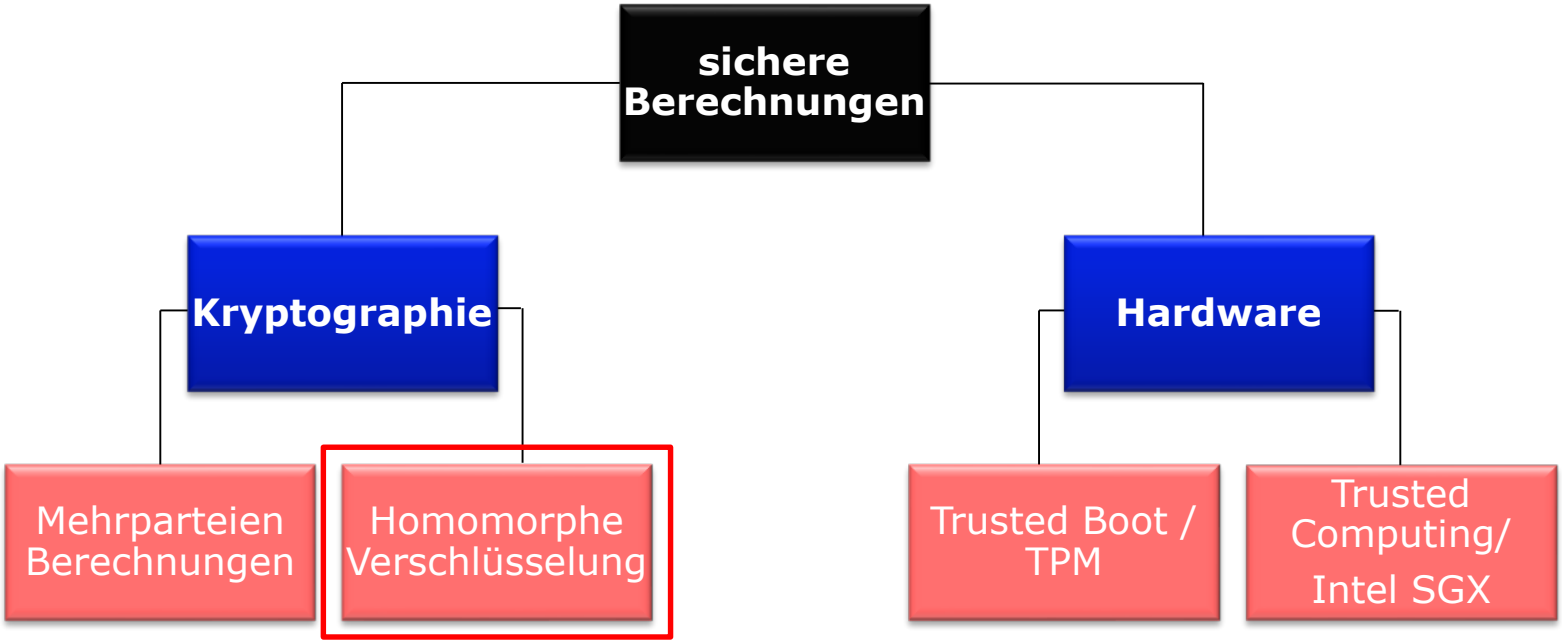


⌘ Praktische Umsetzung:

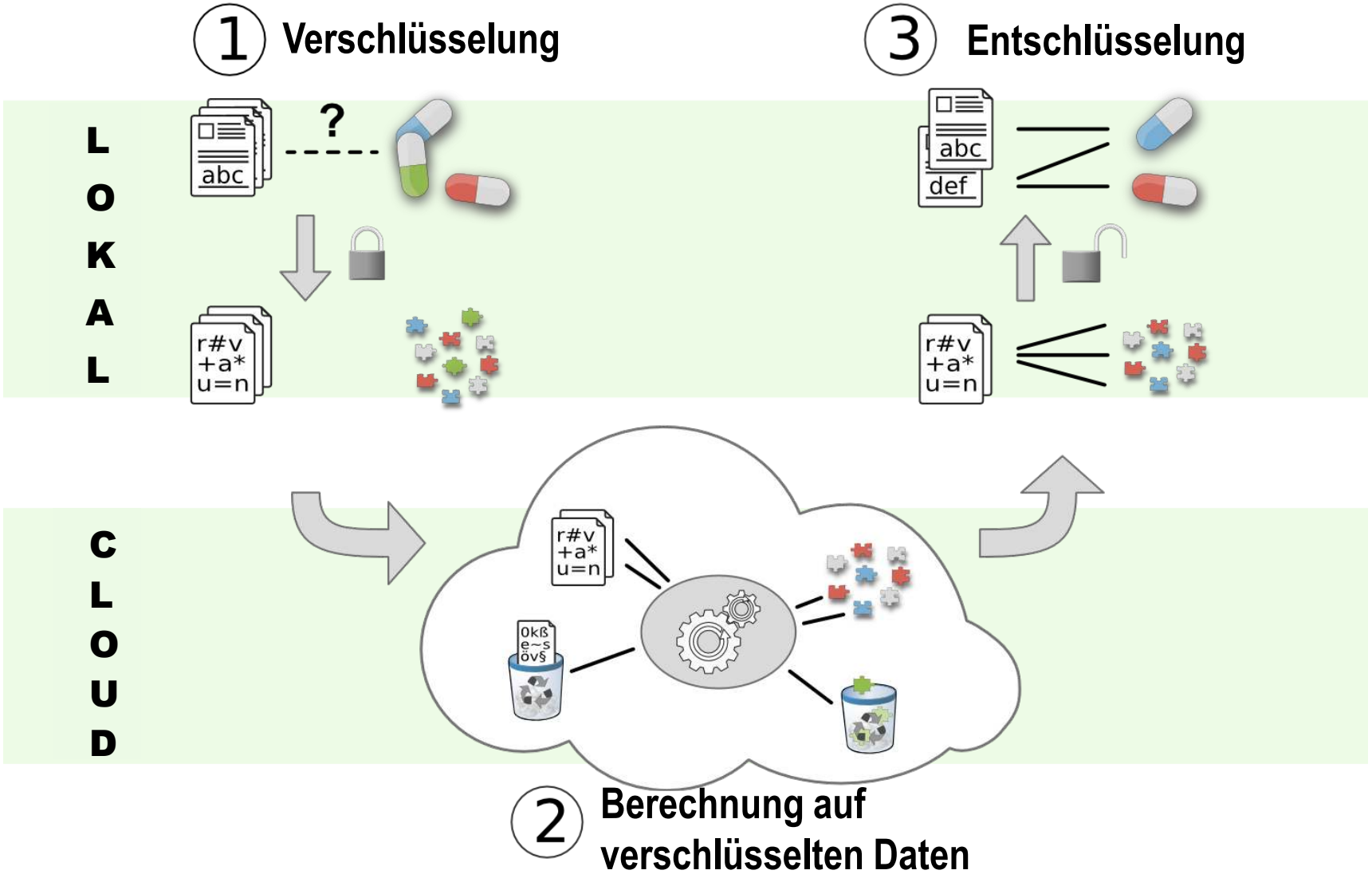
- ⌦ *sharemind* von Cybernetica (Estland)
- ⌦ http://sharemind.cyber.ee/

- Ziel:
  - Auslagerung der Datenverarbeitung
    - **Schutz auch gegenüber dem Betreiber!**

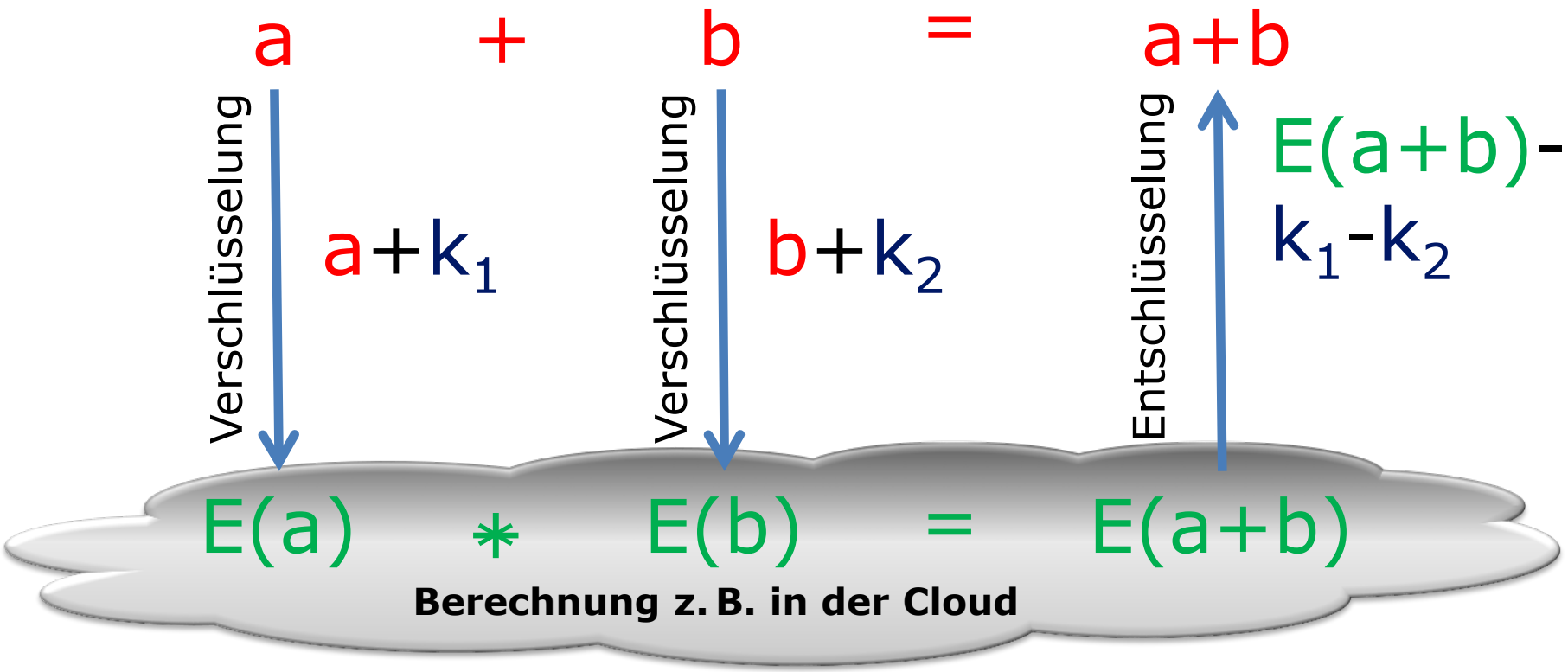
- Lösungsansätze:



Homomorphe Verschlüsselung

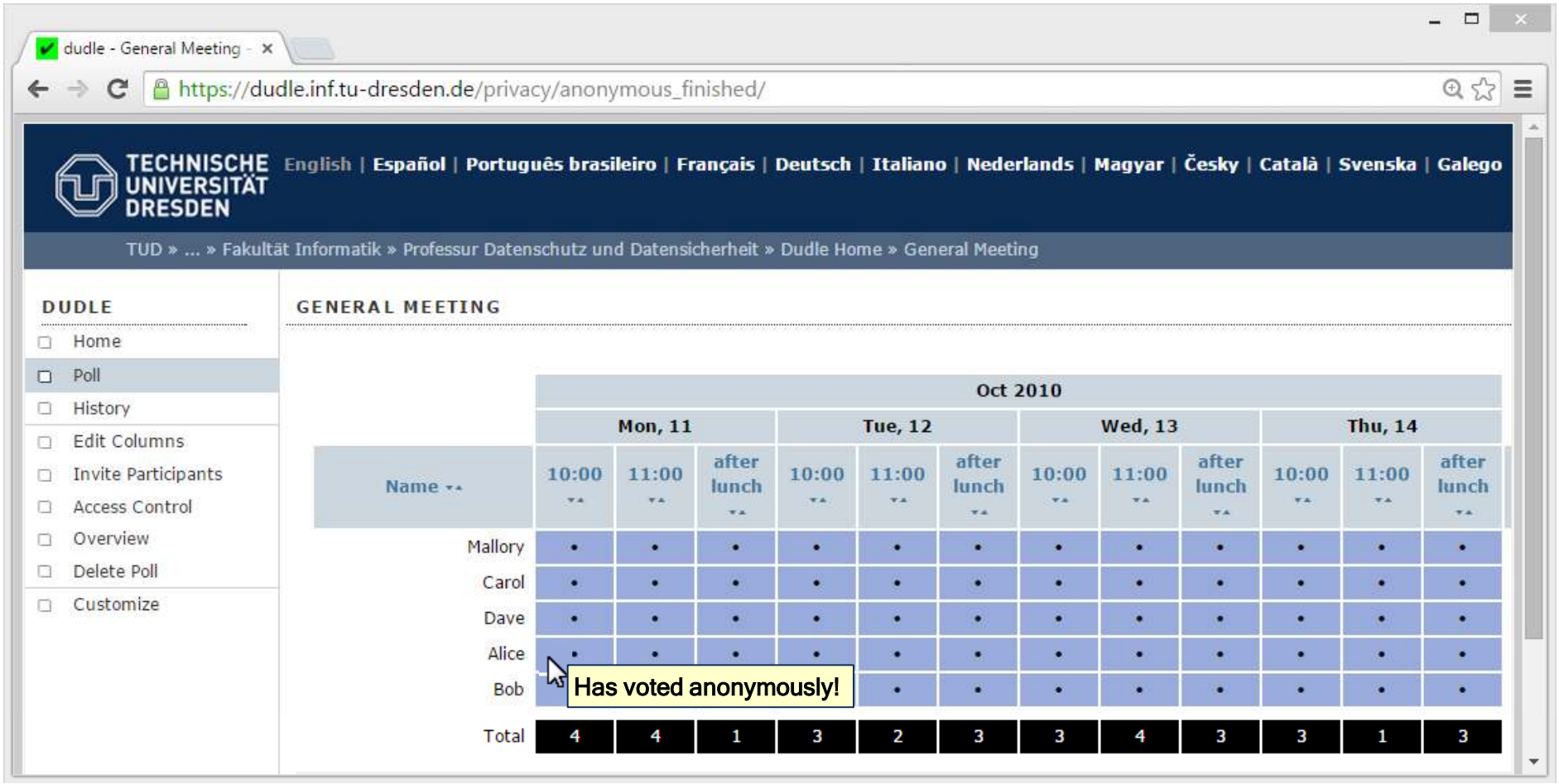


Sichere Berechnungen— Homomorphe Verschlüsselung



- Berechnungen auf verschlüsselten Eingaben
  - ggf. von unterschiedlichen Parteien
- Ausnutzung von Homomorphismus:
  - $f(a) \circ f(b) = f(a+b)$
- im Prinzip: beliebige "Schaltkreise" / Algorithmen berechenbar
  - gewaltiger Rechenaufwand!

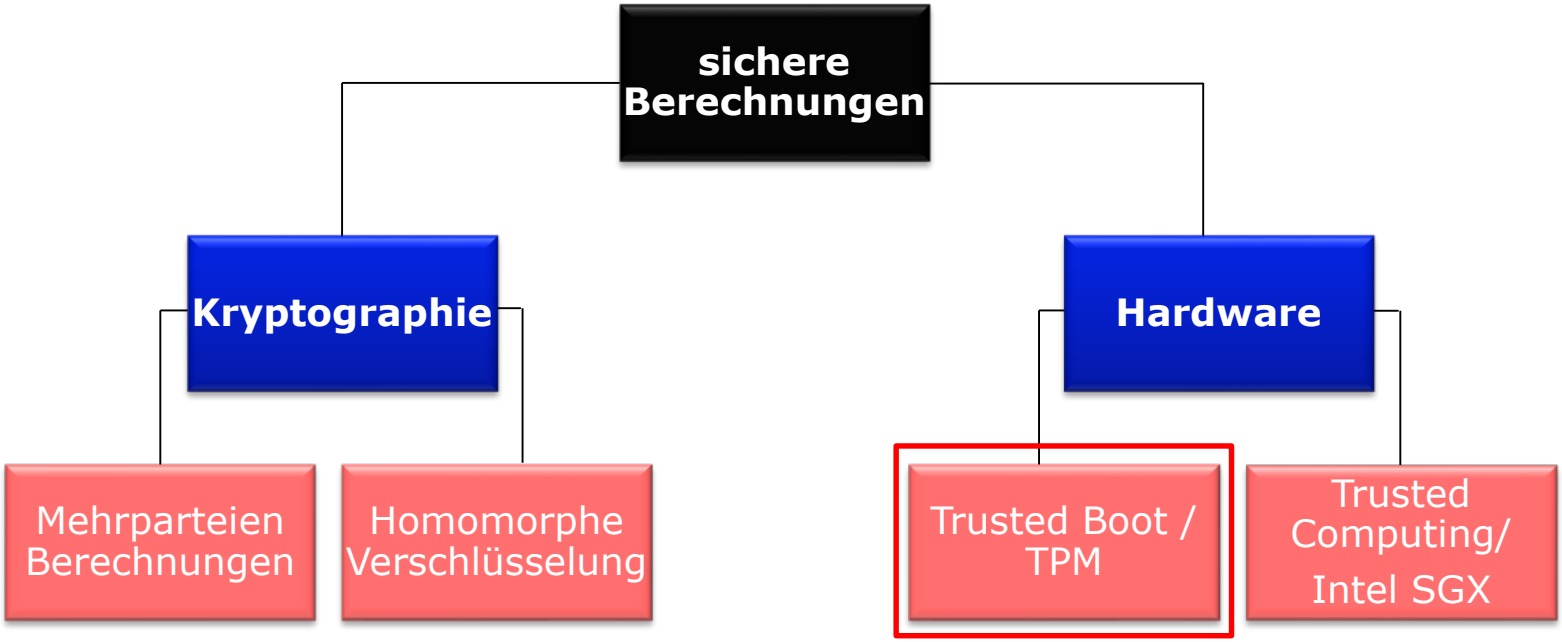
Homomorphe Verschlüsselung— Beispielanwendung: Umfrage



- Termin finden: Homomorphe Addition ohne individuelle Verfügbarkeiten zu verraten
- https://dudle.inf.tu-dresden.de/privacy/

- Ziel:
  - Auslagerung der Datenverarbeitung
    - **Schutz auch gegenüber dem Betreiber!**

- Lösungsansätze:



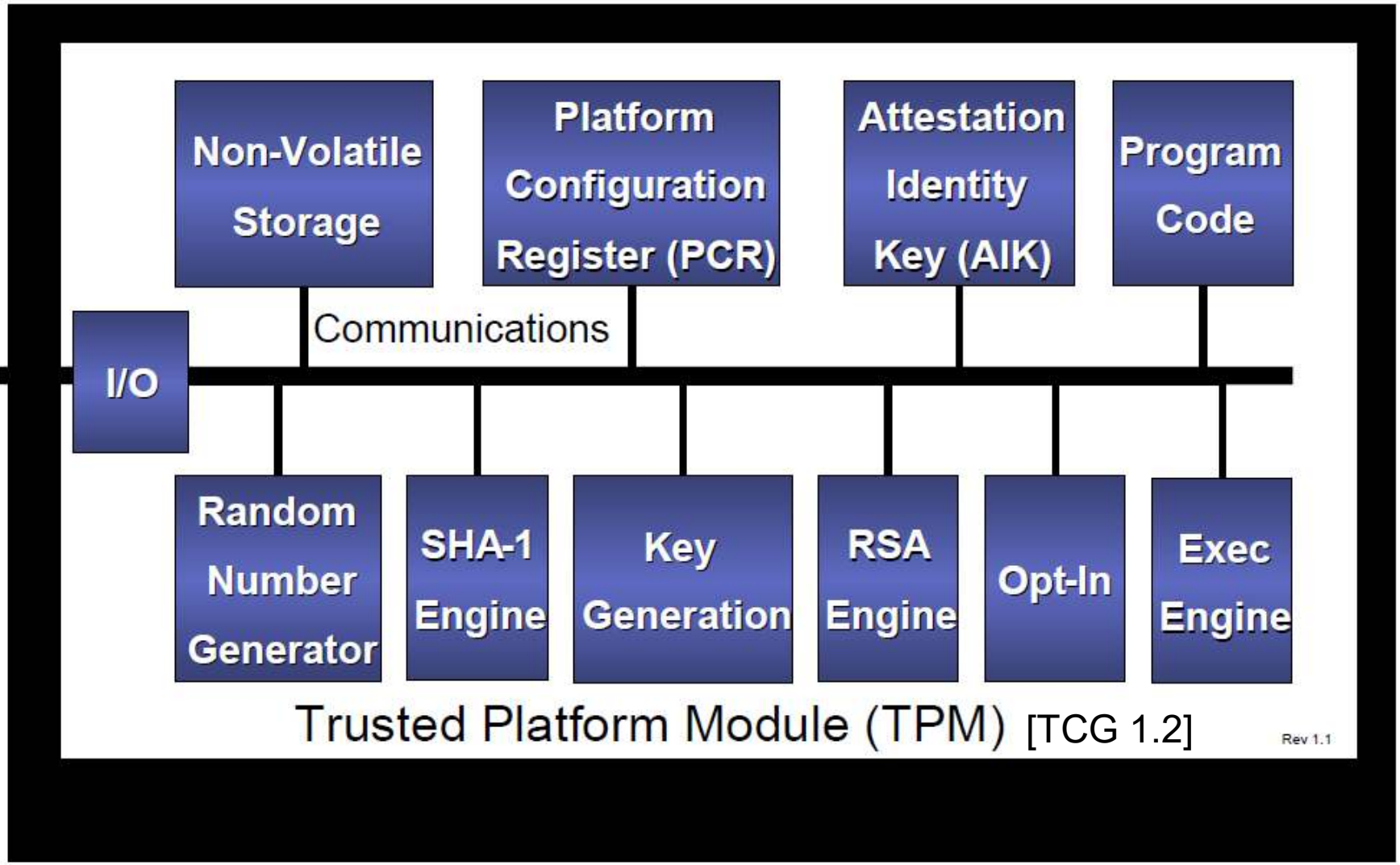
Secure / Authenticated Boot — TPM

- Ziel:
  - sicheres / vertrauenswürdiges Booten
- Secure Boot:
  - nur genau definierte Software-Komponenten werden geladen
- Authenticated Boot:
  - beliebige Software-Komponente werden geladen, aber Systemzustand (remote) überprüfbar
- basiert auf Hardwaresicherheitsmodul (TPM)
  - spezifiziert durch Trusted Computing Group (TCG)
  - http://www.trustedcomputinggroup.org/
  - Bestandteil vieler aktueller Plattformen/Rechner
    - Ggf. proprietäre Lösungen: Apple T2
- A "trusted system or component" is defined as "one which can break the security policy" [Definition des US Department of Defense gemäß Ross Anderson]

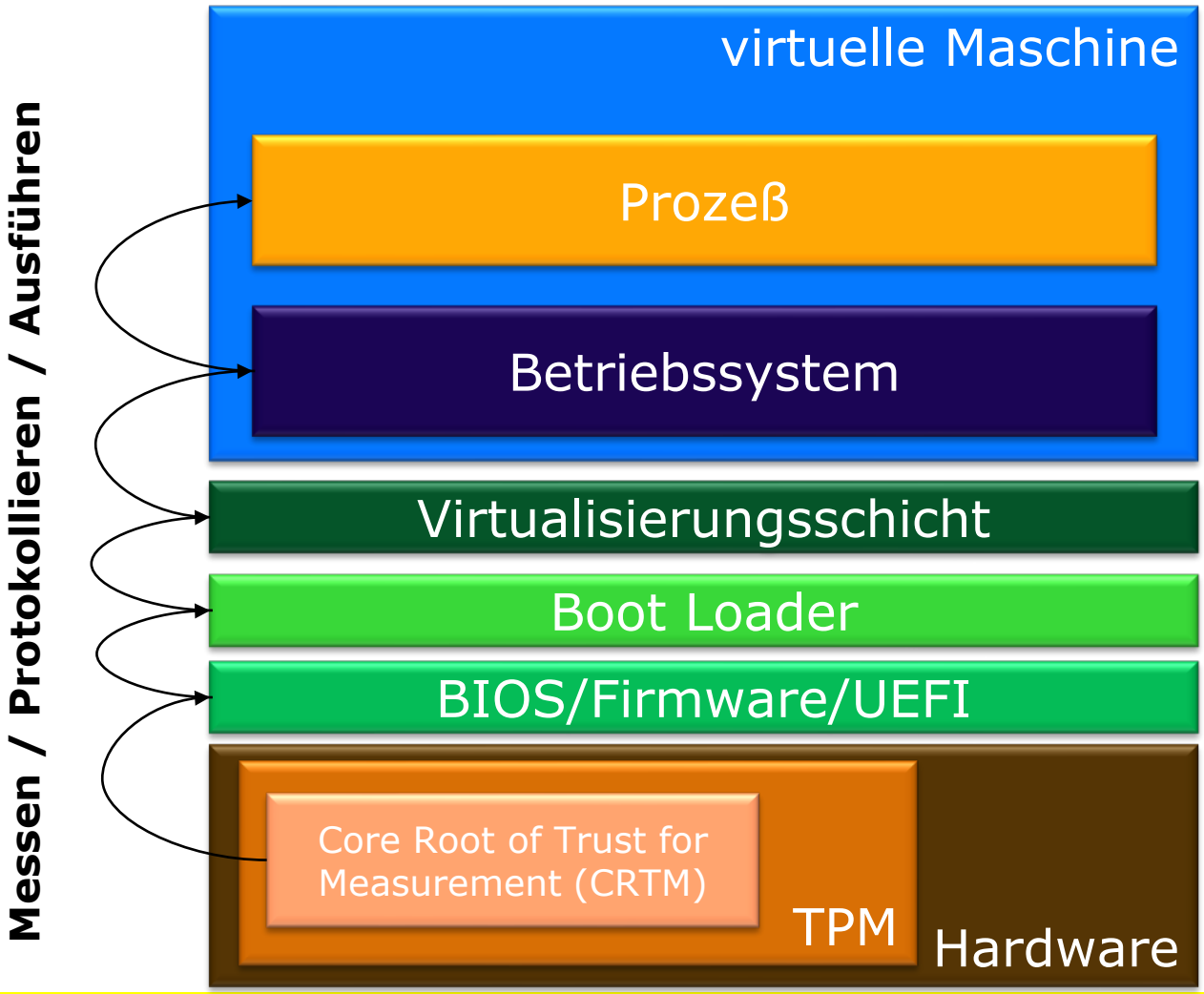
TPM Funktionalitäten

- Sealed storage
  - Verschlüsselung von persistenten Daten
  - Entschlüsselung nur bei gleichem Systemzustand möglich
- Remote attestation
  - Überprüfen des Systemzustands aus der Ferne
- digitale Identität
  - digitale Signatur gebunden an TPM

Trusted Platform Module (TPM)



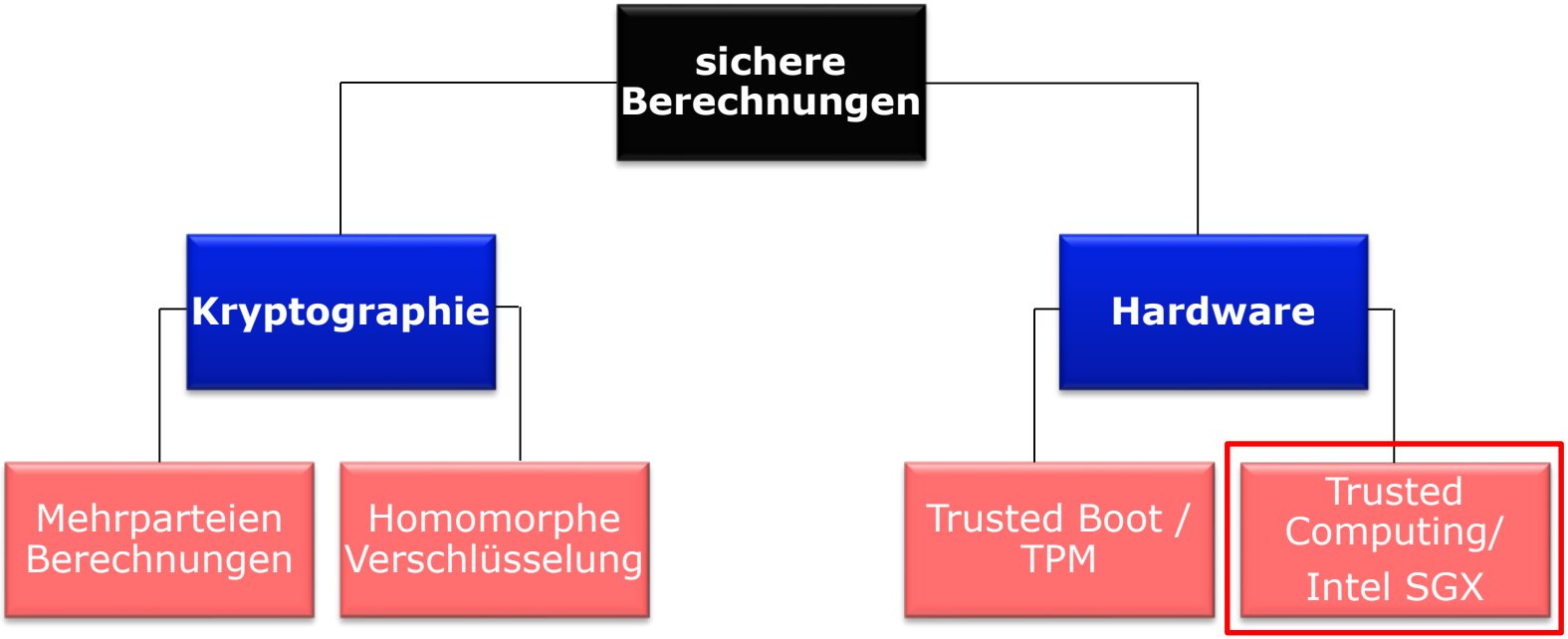
Secure Boot



- Systemzustand überprüfbar
- große Trusted Computing Base (TCB):
  - Hardware, Virtualisierung (Hypervisor), Betriebssystem, Anwendungssoftware
- separate Hardware notwendig (TPM)
- **schwacher** Schutz gegenüber Betreiber der Hardware

- Ziel:
  - Auslagerung der Datenverarbeitung
    - **Schutz auch gegenüber dem Betreiber!**

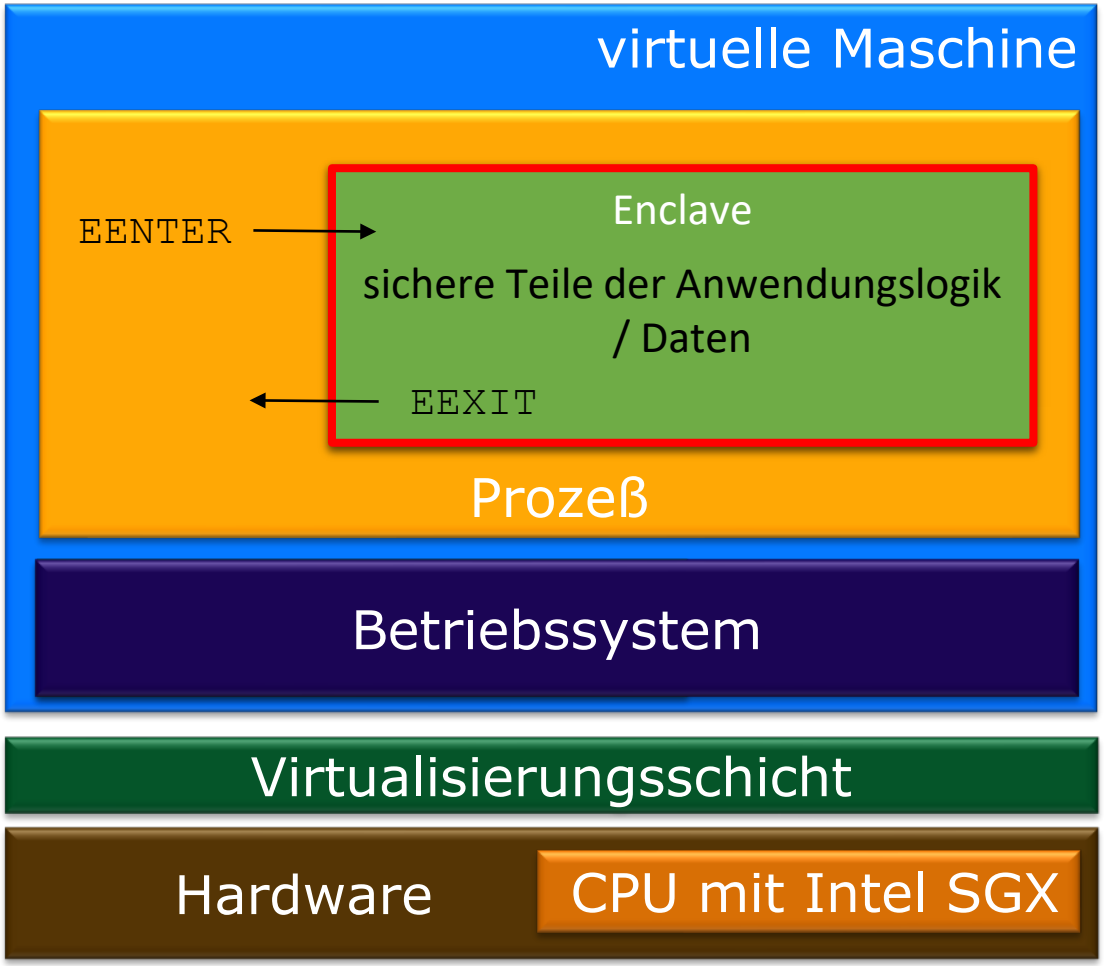
- Lösungsansätze:



- Ziele:
  - sichere Datenverarbeitung auf System, über die man keine physische Kontrolle hat
    - Schaffung einer sicheren Ausführungsumgebung (Trusted Execution Environment)
  - Verringerung der Trusted Computing Base (TCB)

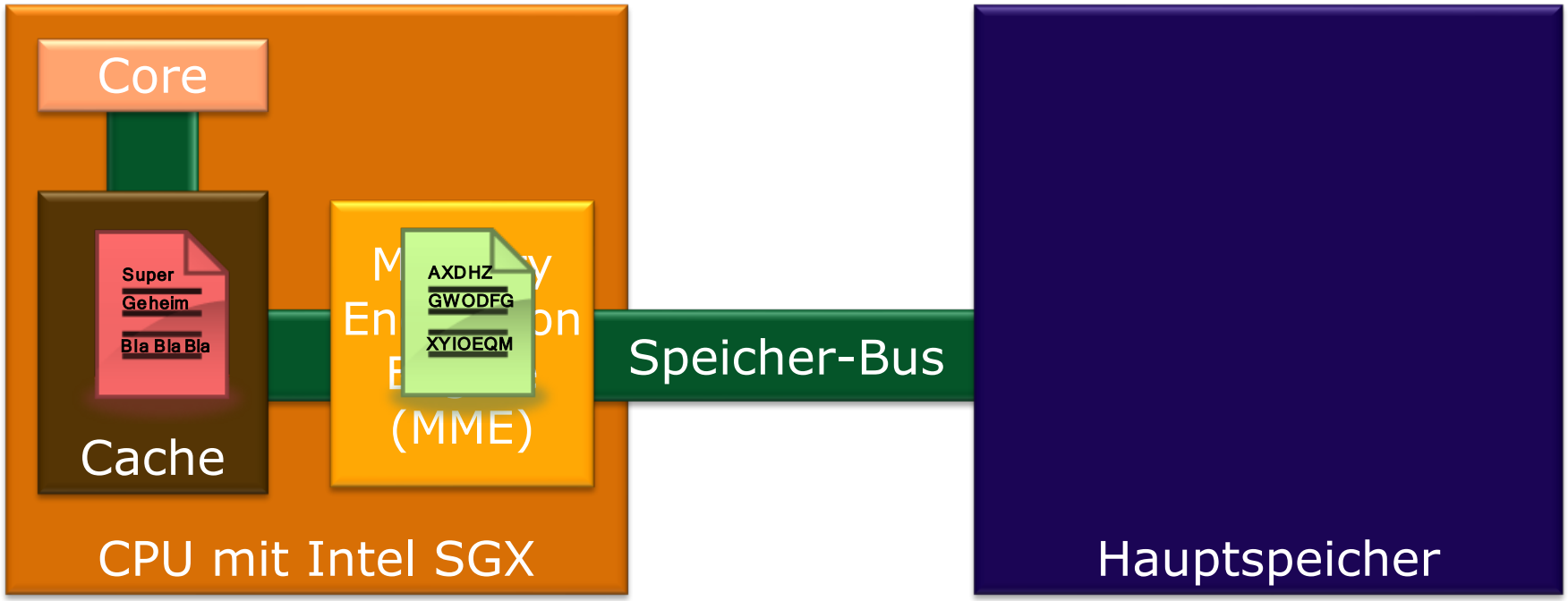
- Lösungsansatz:
  - Hardwaresicherheitsmodul integriert in CPU-Die
  - Programmkode und Daten *außerhalb* der CPU nur *verschlüsselt*
  - *innerhalb* der CPU *unverschlüsselt*
    - Begriff: *Enclave*

Sichere Datenverarbeitung in Enclaven bei Intel SGX



- kleinere Trusted Computing Base (TCB):
  - Hardware (!), Teile der Anwendungs-software
- Schutz gegenüber Betreiber der Hardware
- Remote Attestation von Enclaven möglich
- keine zusätzliche Hardware notwendig
- aktuell: unsichere Umsetzung
  - Seitenkanalangriffe

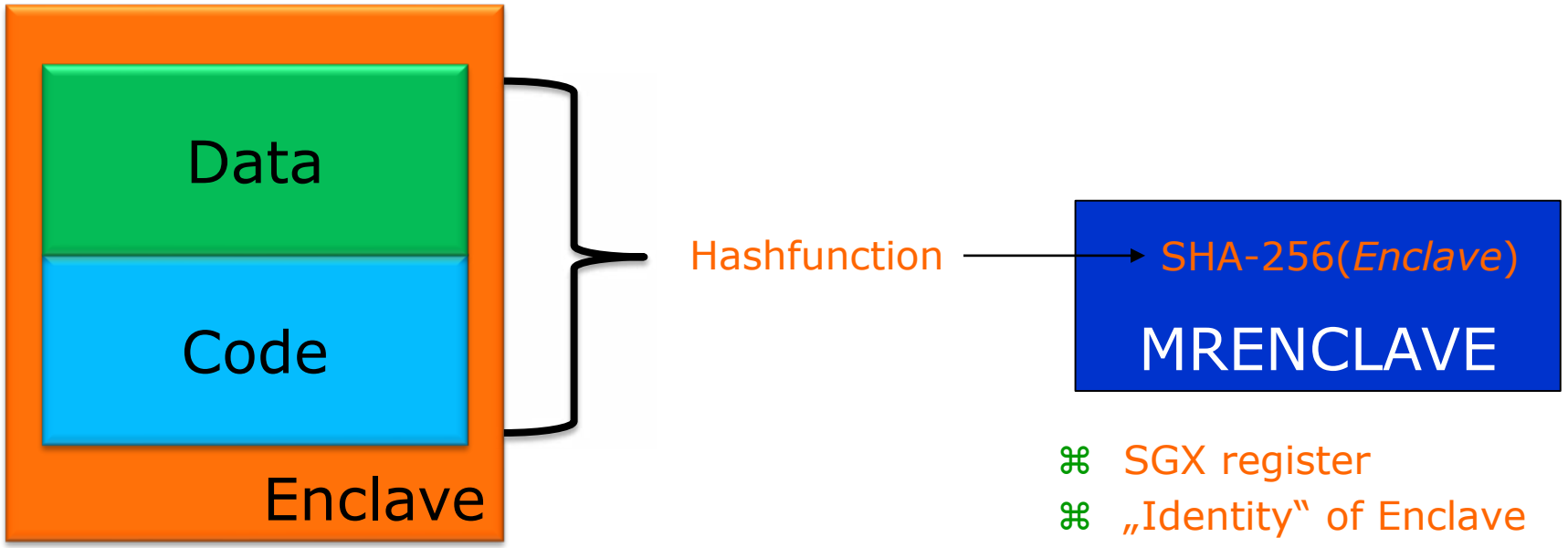
Verschlüsselung zwischen CPU und Hauptspeicher



- innerhalb der CPU: unverschlüsselt
- außerhalb der CPU: verschlüsselt und integritätsgesichert

Some background on cryptography

Enclave Measurement and Identity

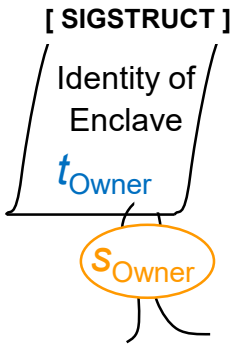


- SGX register
- „Identity“ of Enclave
  - not necessarily unique

- MRENCLAVE is fixed after initialisation of Enclave (EINIT)

Identity of Enclave Owner

- Enclave author/owner/builder has RSA (signature) key pair:
  - public test key: t_{Owner}
  - private signature key: s_{Owner}
- Pseudonym/digital identity of enclave owner: P_{Owner}=SHA-256(t_{Owner})
- Enclave certificate (SIGSTRUCT)
  - checked during Enclave initialisation
  - P_{Owner} is stored in MRSIGNER

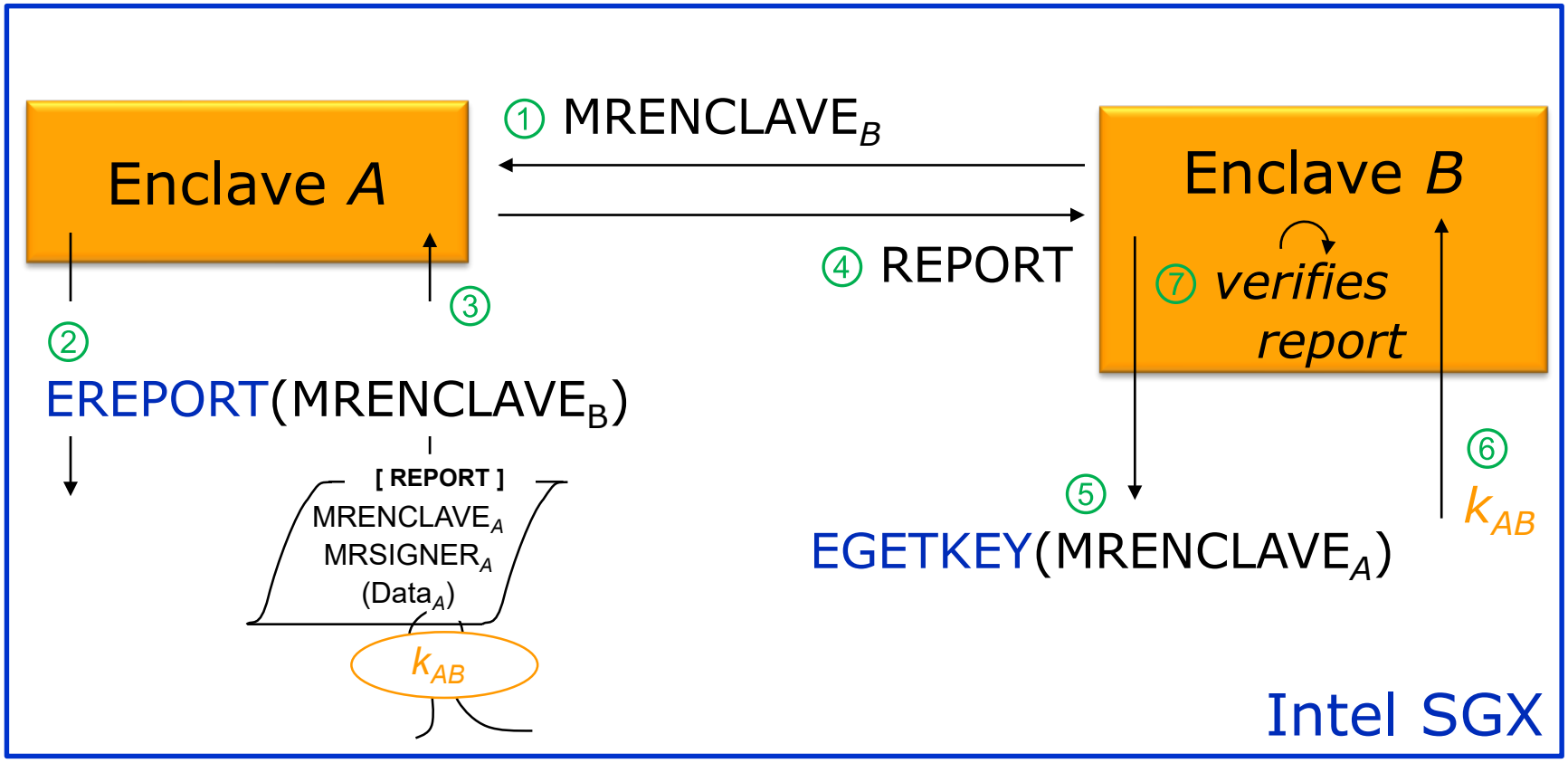


- built-in support for secure data exchange between Enclaves with same P_{Owner}

Enclave Attestation

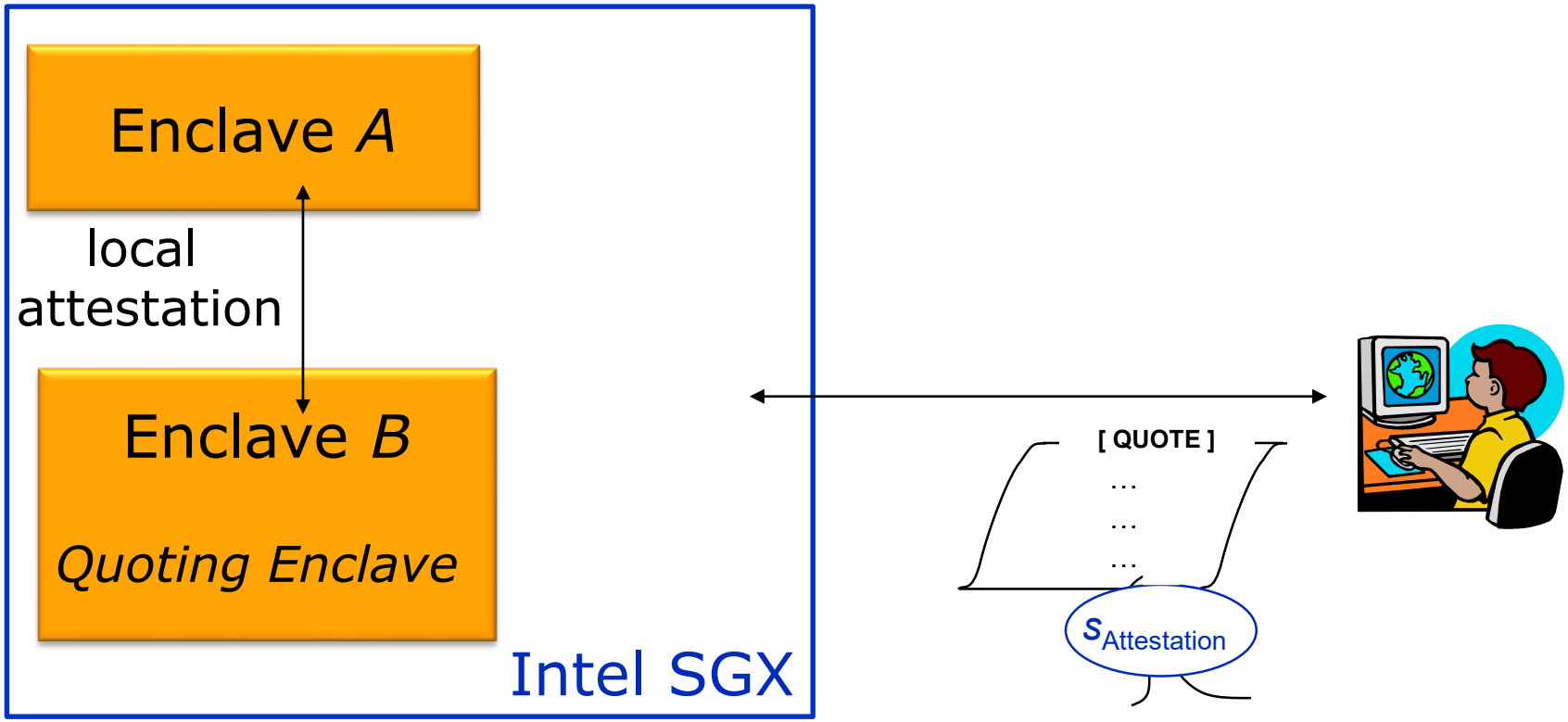
- Intra-Platform Enclave Attestation (*local attestation*)
  - between two Enclaves running on the same hardware
  - allows secure data exchange
- Inter-Platform Enclave Attestation (*remote attestation*)
  - allows an external third party to verify an Enclave

Intra-Platform Enclave Attestation



- Enclave *B* can be sure that Data$_A$ is originated form Enclave *A*
  - use case: Data$_A$ → public key

Inter-Platform Enclave Attestation — simplified view



- Remote attestation is executed by a special Enclave: the Quoting Enclave
- signature scheme is a group signature
  - Intel Enhanced Privacy ID (EPID)
  - $s_{Attestation}$ is unique per device

Group Signatures
(Chaum, van Heyst 1991)

- Idea: digital signature on behalf of a group without revealing which group member did sign
- Setting:
  - Group Manager (can be distributed):
    - generates group key pair
    - join / leave of group members
    - revoke anonymity of group members
  - Join:
    - member learns **his** private key for signing
  - Leave:
    - private key of the member is revoked
  - Signing:
    - every member of group
  - Verification:
    - everybody with the help of the group public key

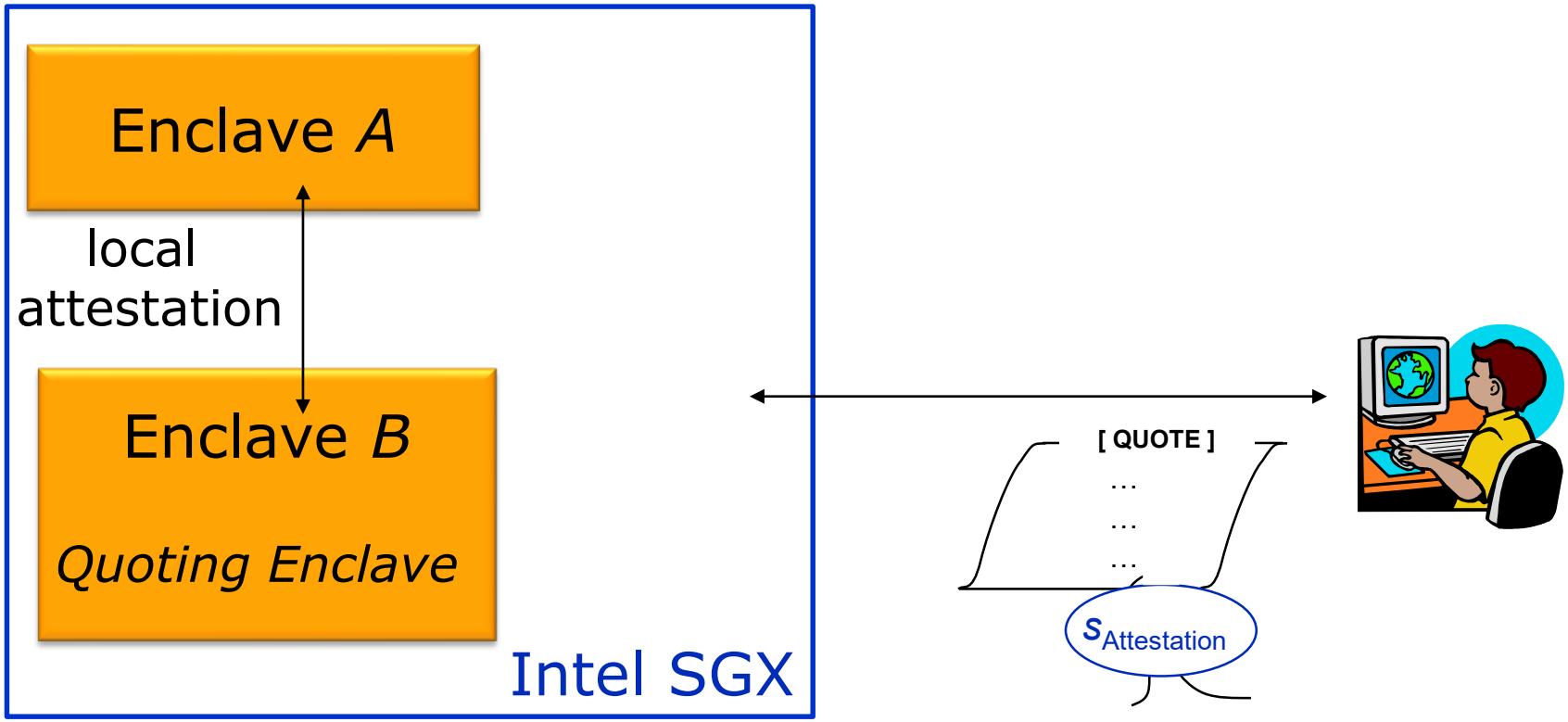
Properties of a Group Signature Scheme

- Soundness and Completeness
  - valid signatures always verify correctly
  - invalid signatures always fail verification.
- Unforgeable
  - only group members can create valid signatures
- Anonymity
  - given a message and its signature, the signing group member cannot be determined (sometimes: without the group manager's private key)
- Traceability
  - group manager can trace which group member issued a signature
- Unlinkability
  - given two messages and their signatures, only group manager can tell if the signatures were from the same signer or not

Properties of a Group Signature Scheme

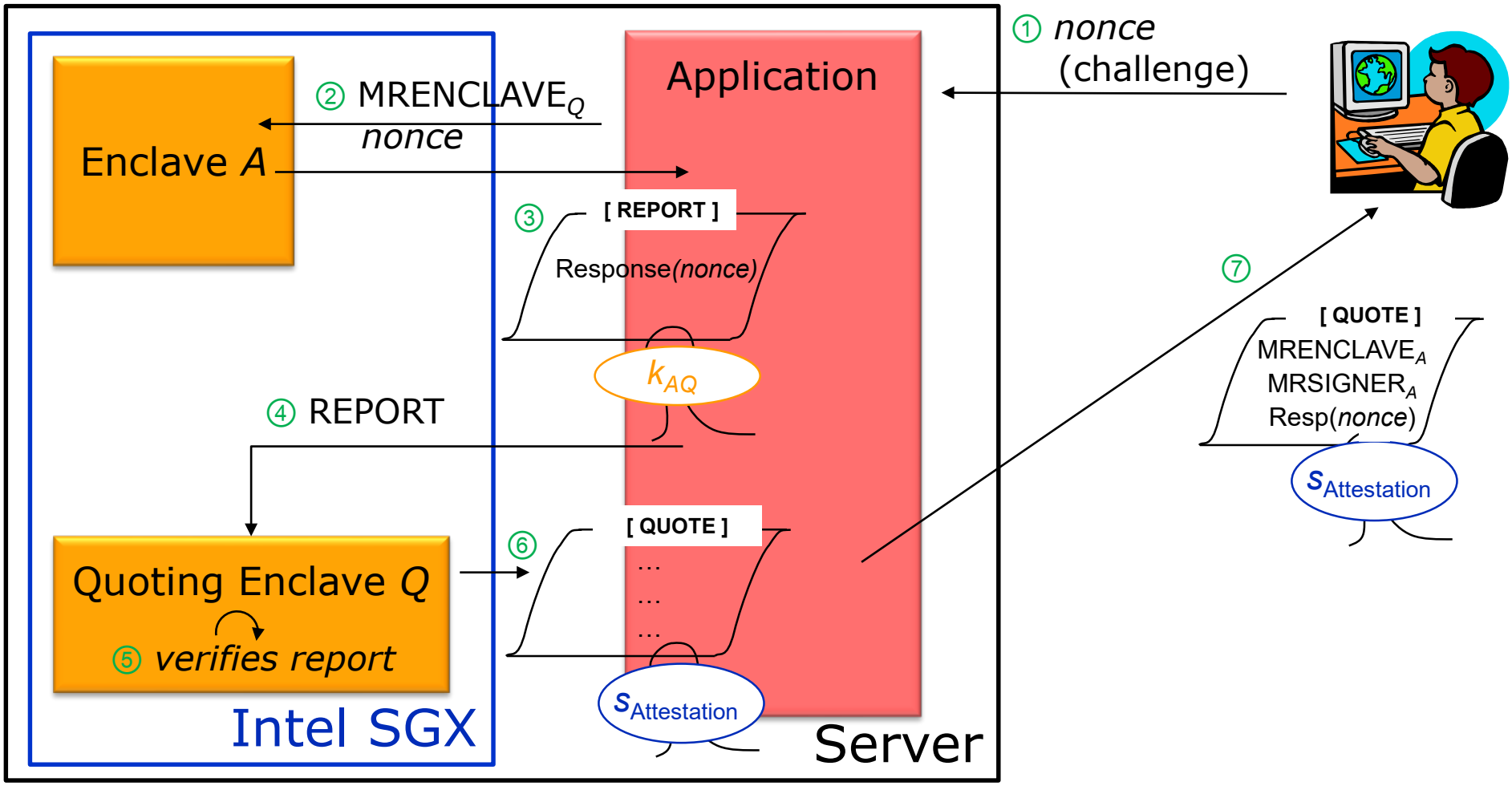
- No Framing
  - colluding group members (and manager) cannot forge a signature of a non-participating group member
- Unforgeable tracing verification
  - group manager cannot falsely accuse a signer of creating a signature he did not create
- Coalition resistance
  - colluding group members cannot generate a signature that the group manager cannot trace to one of the colluding group members

Inter-Platform Enclave Attestation — simplified view



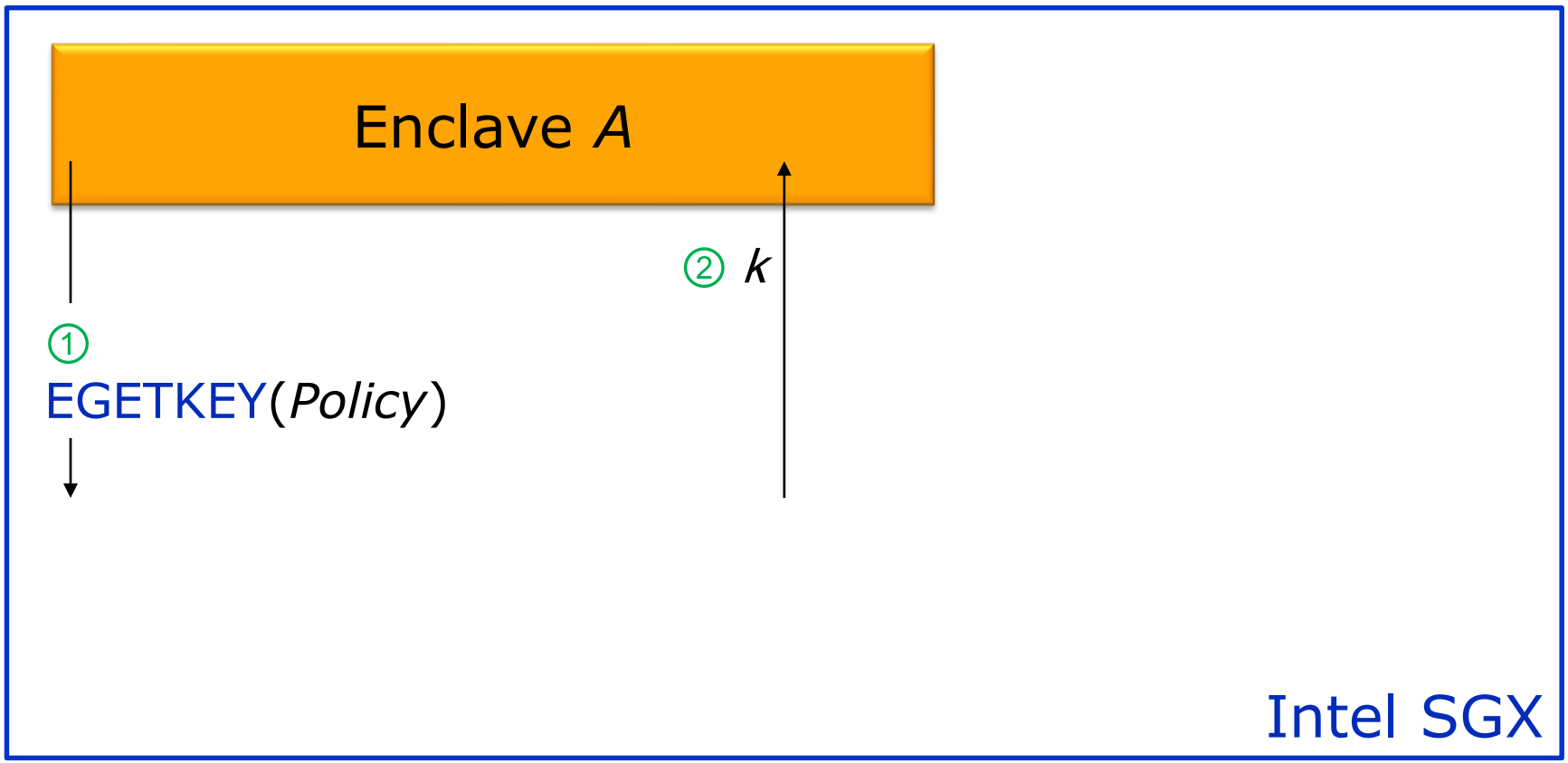
- Remote attestation is executed by a special Enclave: the Quoting Enclave
- signature scheme is a group signature
  - Intel Enhanced Privacy ID (EPID)
  - $s_{Attestation}$ is unique per device

Inter-Platform Enclave Attestation — enhanced view



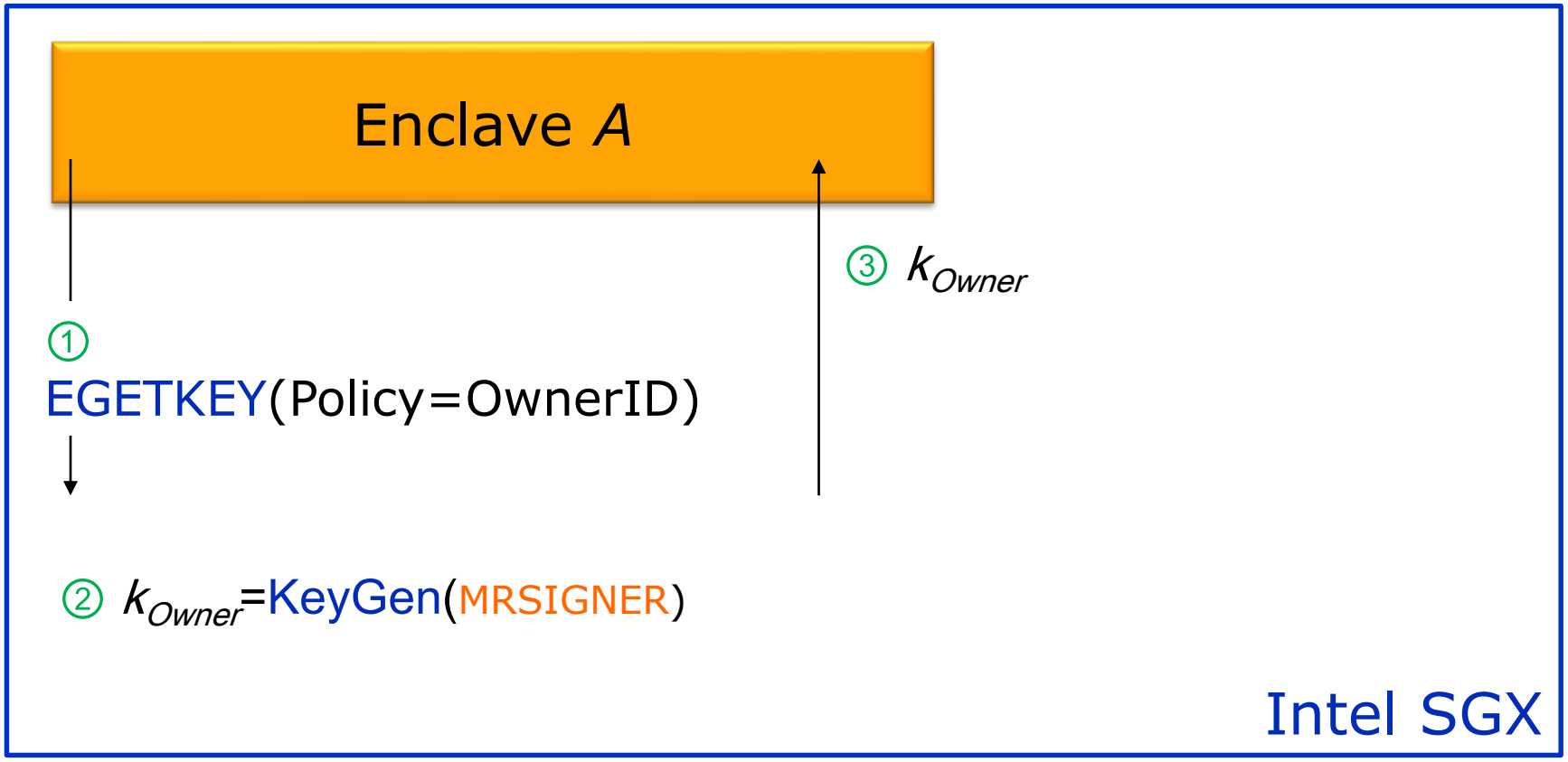
- *nonce* ensures freshness
- testkey $t_{Attestation}$ encodes the SGX (firmware) version (Trusted Computing Base)

Enclave Data Sealing (encryption)



⌘ it's only about the keys → not the encryption itself

Sealing to the Enclave Owner Identity—simplified view



⌘ KeyGen(MRSIGNER)=KeyGen(MRSIGNER') ←→MRSIGNER=MRSIGNER'

Sealing to the Enclave Owner Identity—enhanced view

Enclave *A*

① EGETKEY(Policy=OwnerID, minVersion)

② if(*Version* ($Enclave_A$) < minVersion)
k_{Owner} = NULL
else
k_{Owner}=KeyGen(MRSIGNER, minVersion)

③ k_{Owner}

Intel SGX

⌘ older software cannot access data of newer software

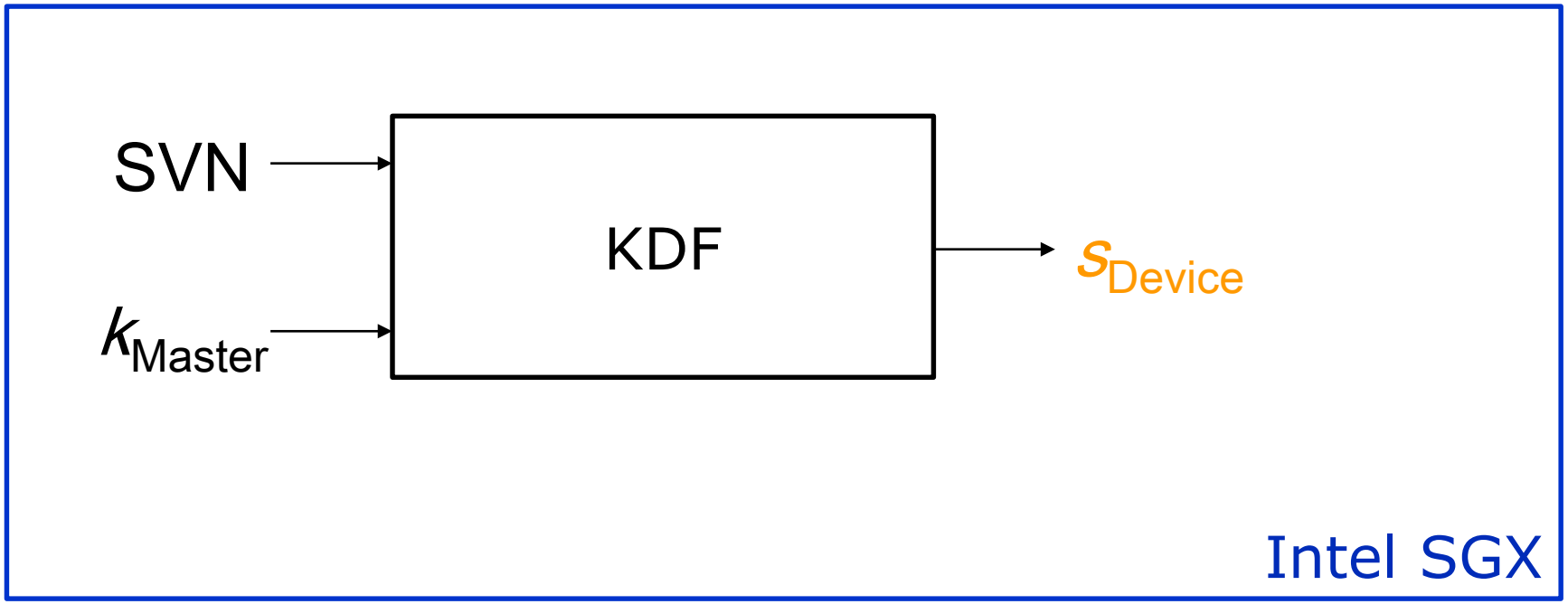
Mysterious OwnerEpoch

- EGETKEY(…,OwnerEpoch)
  - OwnerEpoch
    - is kind of master secret
    - can be changed by the platform owner
      - use case: permanent/temporarily access prevention to keys

Derivation of Attestation Keys (s_{Device}, $s_{Attestation}$)

- s_{Device} depends on Trusted Computing Base (TCB)
  - SGX hardware revision
  - SGX software revision

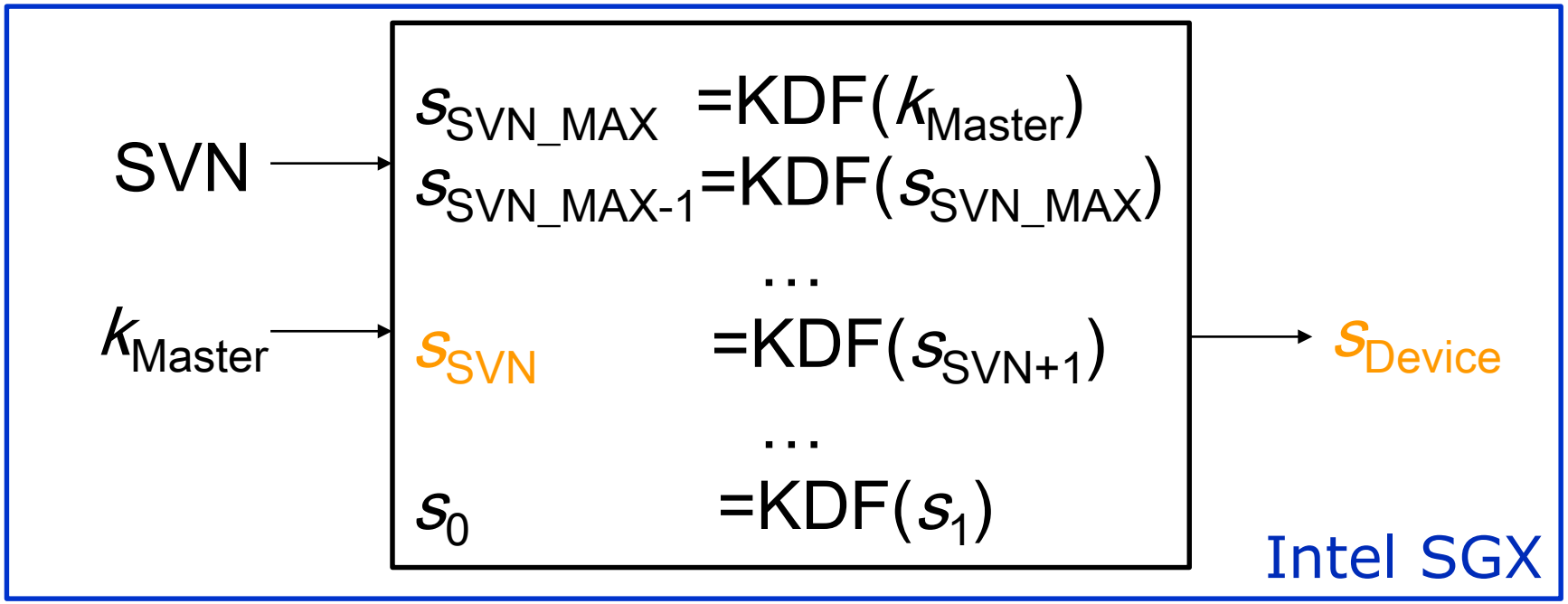
- each revision of the TCB has an increasing Security Version Number (SVN)



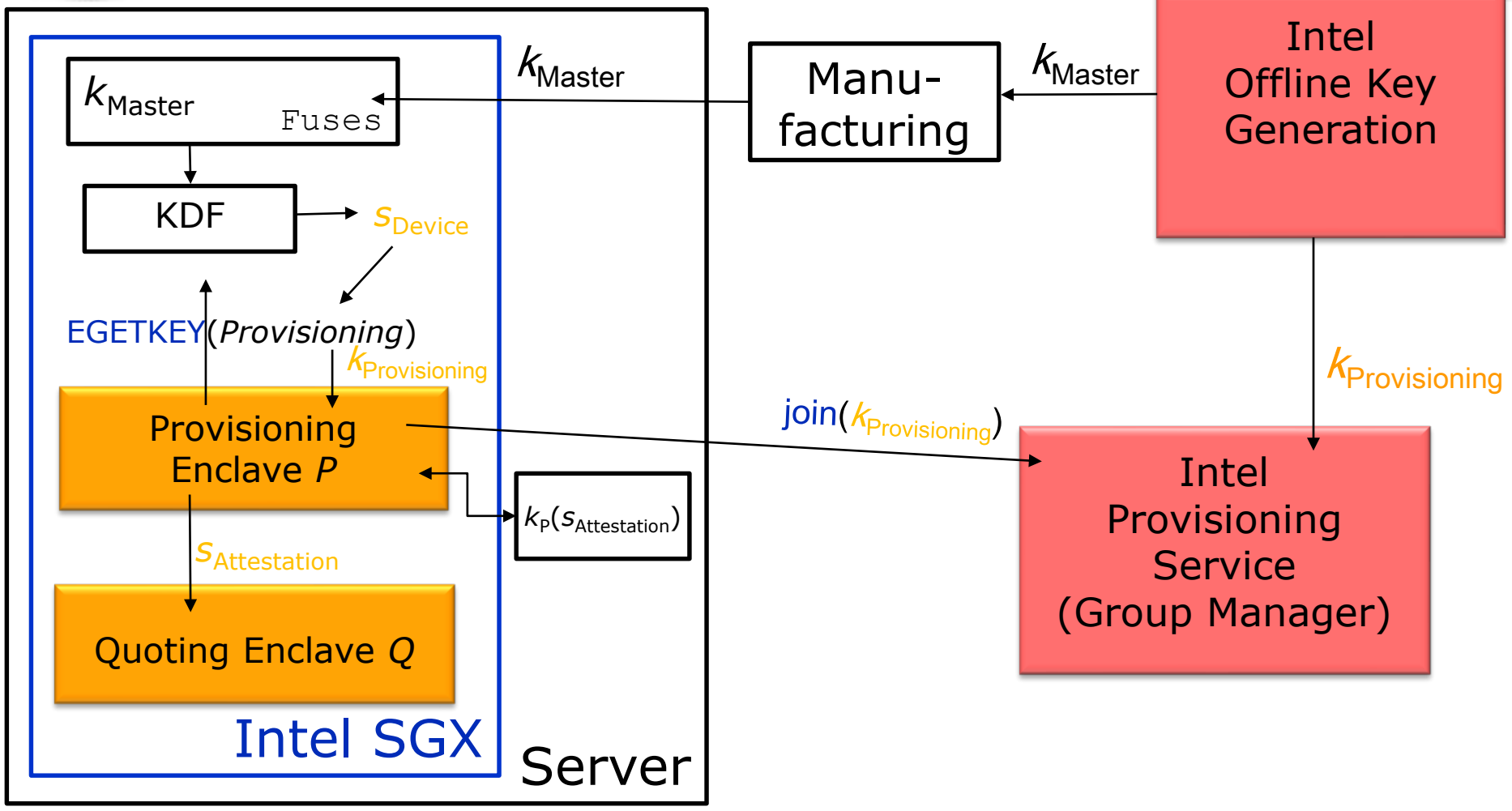
Derivation of Attestation Keys (s_{Device})

- s_{Device} depends on Trusted Computing Base (TCB)
  - SGX hardware revision
  - SGX software revision

- each revision of the TCB has an increasing Security Version Number (SVN)

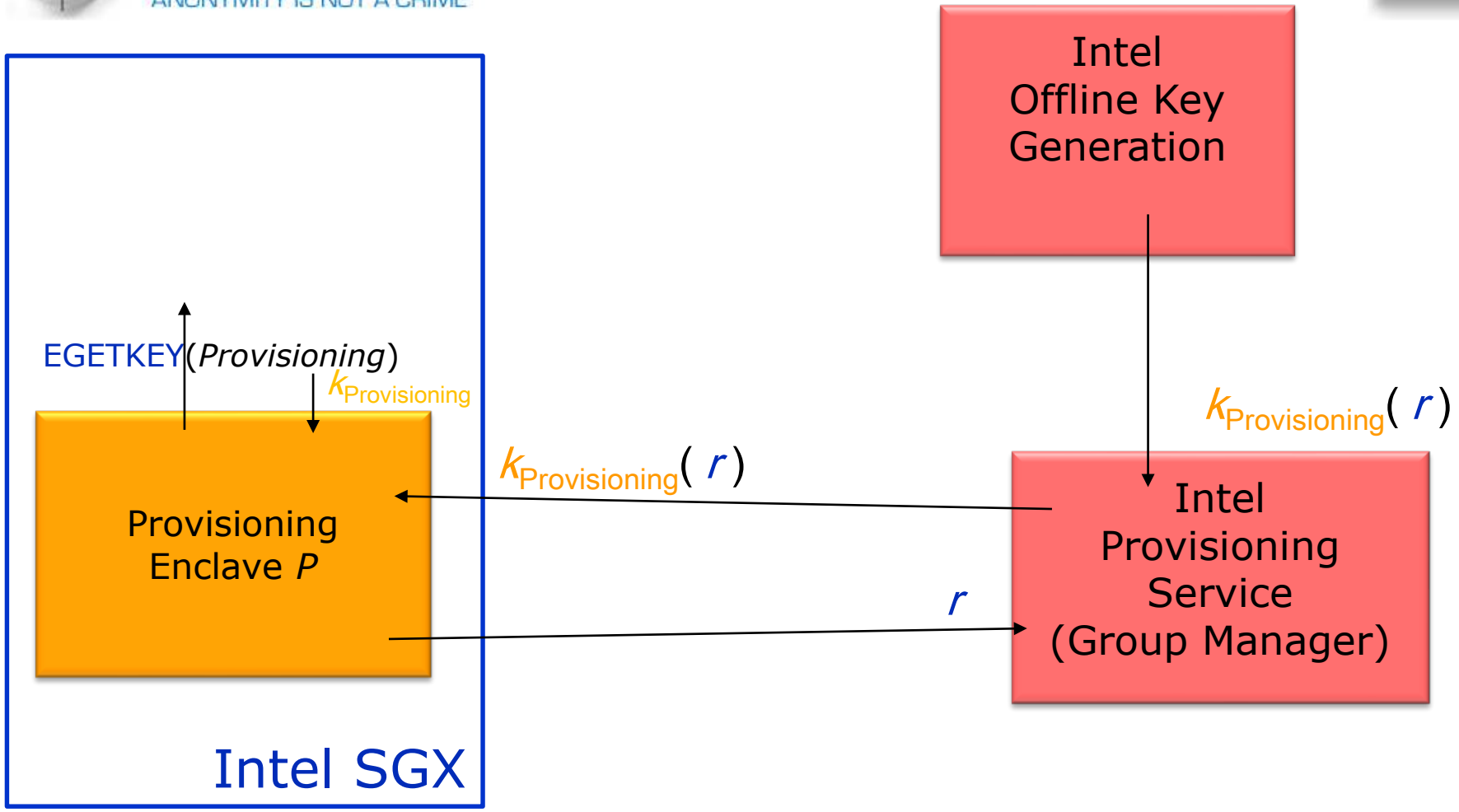


Master Key Creation and Attestation Key Registration (with the Group Manager)



⌘ join will ***not*** reveal $s_{Attestation}$ to the Group manager

Prove of TCB during join() — Method 1: Challenge-Response



Prove of TCB during join() — Method 2: Signature-based

Intel Offline Key Generation

t_{TCB}=KeyGen$_{EC\text{-}DSA}$ ($k_{Provisioning}$)

EGETKEY(*Provisioning*)

$k_{Provisioning}$

Provisioning Enclave *P*

s_{TCB}=KeyGen$_{EC\text{-}DSA}$ ($k_{Provisioning}$)

nonce

s_{TCB}(*nonce*)

Intel Provisioning Service (Group Manager)

Intel SGX

Intel SGX: aktueller Stand

- Hardware verfügbar
- SDK (Windows, Linux) verfügbar
  - Umfangreiche Anpassungen / Neu-Schreiben von Anwendung notwendig
- Forschungs-Prototypen:
  - Microsoft Haven
    - komplette VM innerhalb Enclave
  - Microsoft VC3
    - Map-Reduce mit Hadoop
    - nur Map- und Reduce-Funktionen in Enclave
- Forschungsprojekte mit TUD-Beteiligung
  - Secure Enclaves for REactive Cloud Applications (SERECA)
  - Secure Big Data Processing in Untrusted Clouds (SecureCloud)
  - Ergebnis: SCONE: Plattform für Cross-Compilierung von Anwendungen zur Ausführung in Enclave (scontain.com)

SERECA

SecureCloud