Westfälische Hochschule
Gelsenkirchen Bocholt Recklinghausen
University of Applied Sciences

Trusted Computing

- Vorlesung Cyber-Sicherheit -

Prof. Dr. (TU NN)
Norbert Pohlmann

Institut für Internet-Sicherheit – if(is)
Westfälische Hochschule, Gelsenkirchen
http://www.internet-sicherheit.de

Trusted Computing
→ Inhalt

- Ziele und Ergebnisse der Vorlesung
- Kernfunktionalitäten
- Sicherheitsarchitektur
- Trusted Network Connect
- Zusammenfassung

Trusted Computing
→ Inhalt

- **Ziele und Ergebnisse der Vorlesung**
- Kernfunktionalitäten
- Sicherheitsarchitektur
- Trusted Network Connect
- Zusammenfassung

Ziele und Ergebnisse der Vorlesung
→ Trusted Computing

- Gutes Verständnis über die **Sicherheitsarchitektur** und **Sicherheitsprinzipien** von Trusted Computing erlangen.
- Erlangen der Kenntnisse über die TPM Schlüsselhierarchie, Authenticated Boot, Attestation, Binding, Sealing und Trusted Network Connect.
- Gutes Verständnis zu verschiedenen **Kernelarchitekturen** erlangen.
- Gutes Verständnis über praktische Anwendungen durch Betrachtung von **Turaya** als Beispielanwendung.

Trusted Computing
→ Inhalt

- Ziele und Ergebnisse der Vorlesung
- **Kernfunktionalitäten**
- Sicherheitsarchitektur
- Trusted Network Connect
- Zusammenfassung

Kernfunktionalitäten
→ Herausforderungen

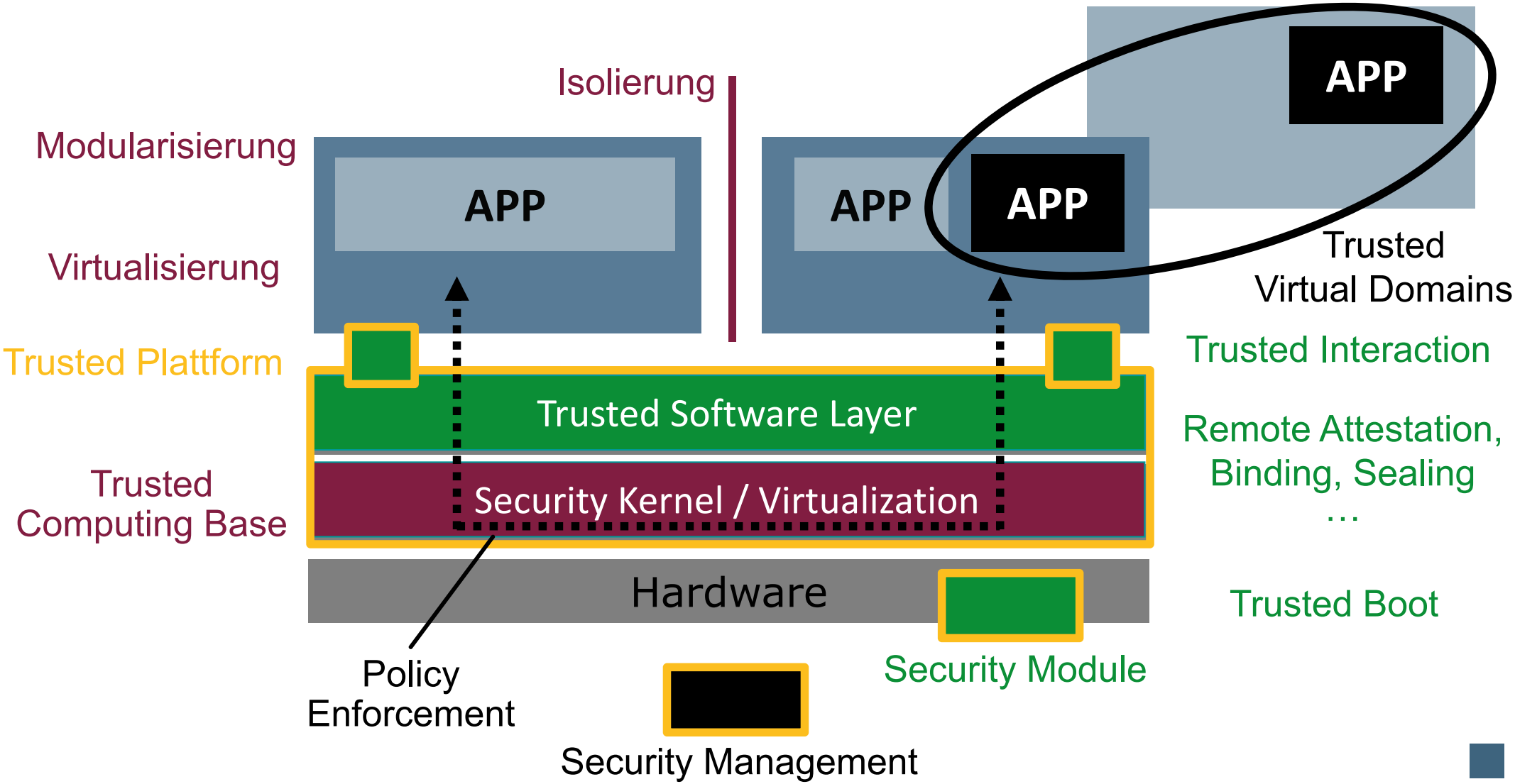
- **Reaktive Cyber-Sicherheitssysteme**
  - „Airbag-Methode“

- **Proaktive Cyber-Sicherheitssysteme**
  - „ESP-Strategie“

- **Das Software-Problem**

Kernfunktionalitäten
→ Sicherheitsprinzipien

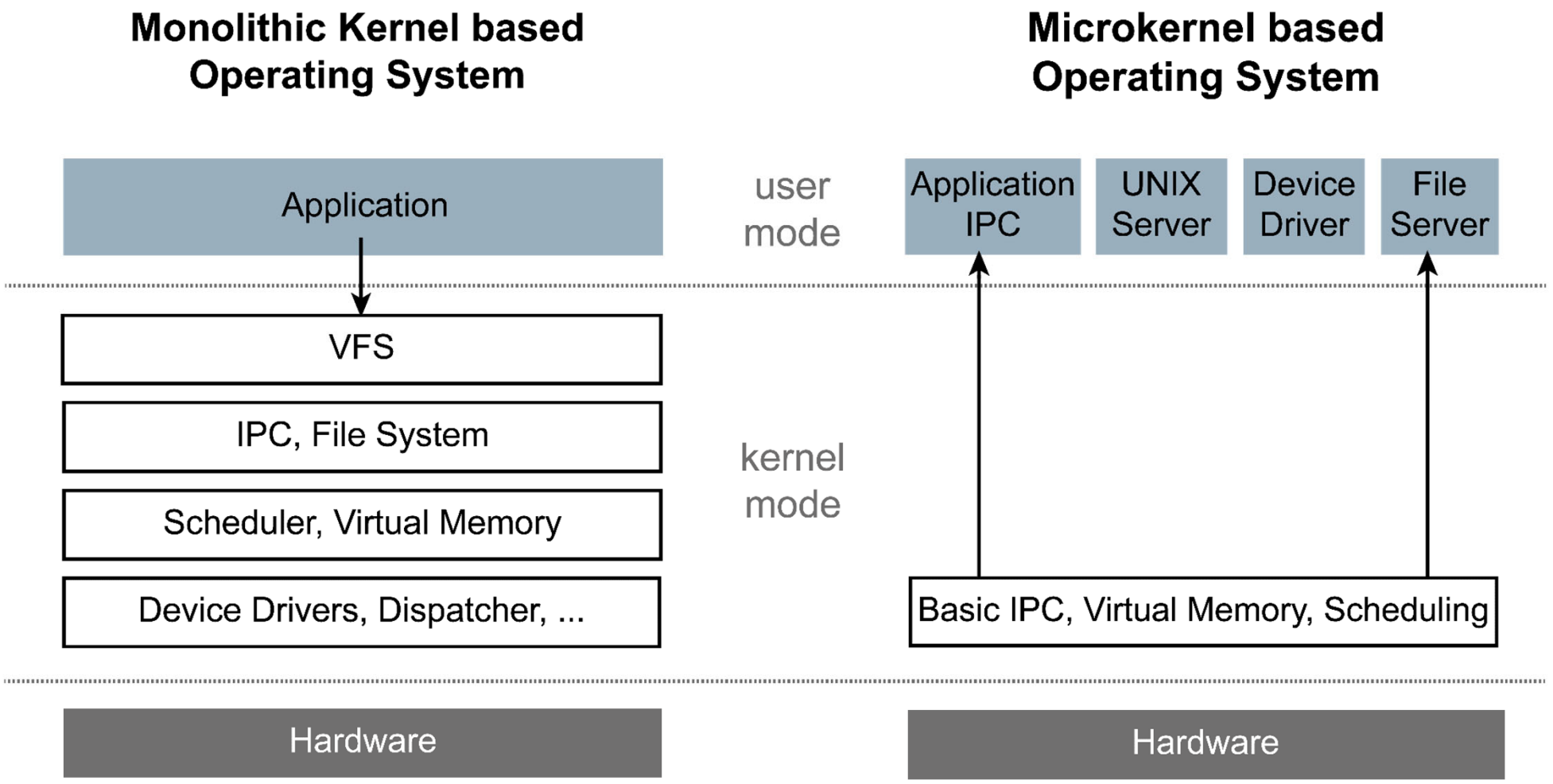
Robustness/Modularity **Trusted Process** **Integritätsprüfung**



Trusted Computing
→ Inhalt

- Ziele und Ergebnisse der Vorlesung
- Kernfunktionalitäten
- **Sicherheitsarchitektur**
- Trusted Network Connect
- Zusammenfassung

Sicherheitsarchitektur
→ Kernelarchitekturen (1/3)



Sicherheitsarchitektur
→ Kernelarchitekturen (2/3)

- **Vorteile** eines monolithischen Kernels:
  - Lange etabliert
  - Gute Performance

- **Nachteile** eines monolithischen Kernels:
  - Alle Treiber vereint im Kernel-Space
  - Geringere Flexibilität
  - Höhere Komplexität
  - Wenig robust
  - Schlechte Sicherheitsmechanismen

Sicherheitsarchitektur
→ Kernelarchitekturen (3/3)

- **Vorteile** eines Mikrokernels:
  - Höhere Robustheit
  - Höhere Modularität
  - Höhere Flexibilität
  - Höhere IT-Sicherheit (kontrollierbare Interprozesskommunikation)
  - Weniger benötigter Speicherplatz

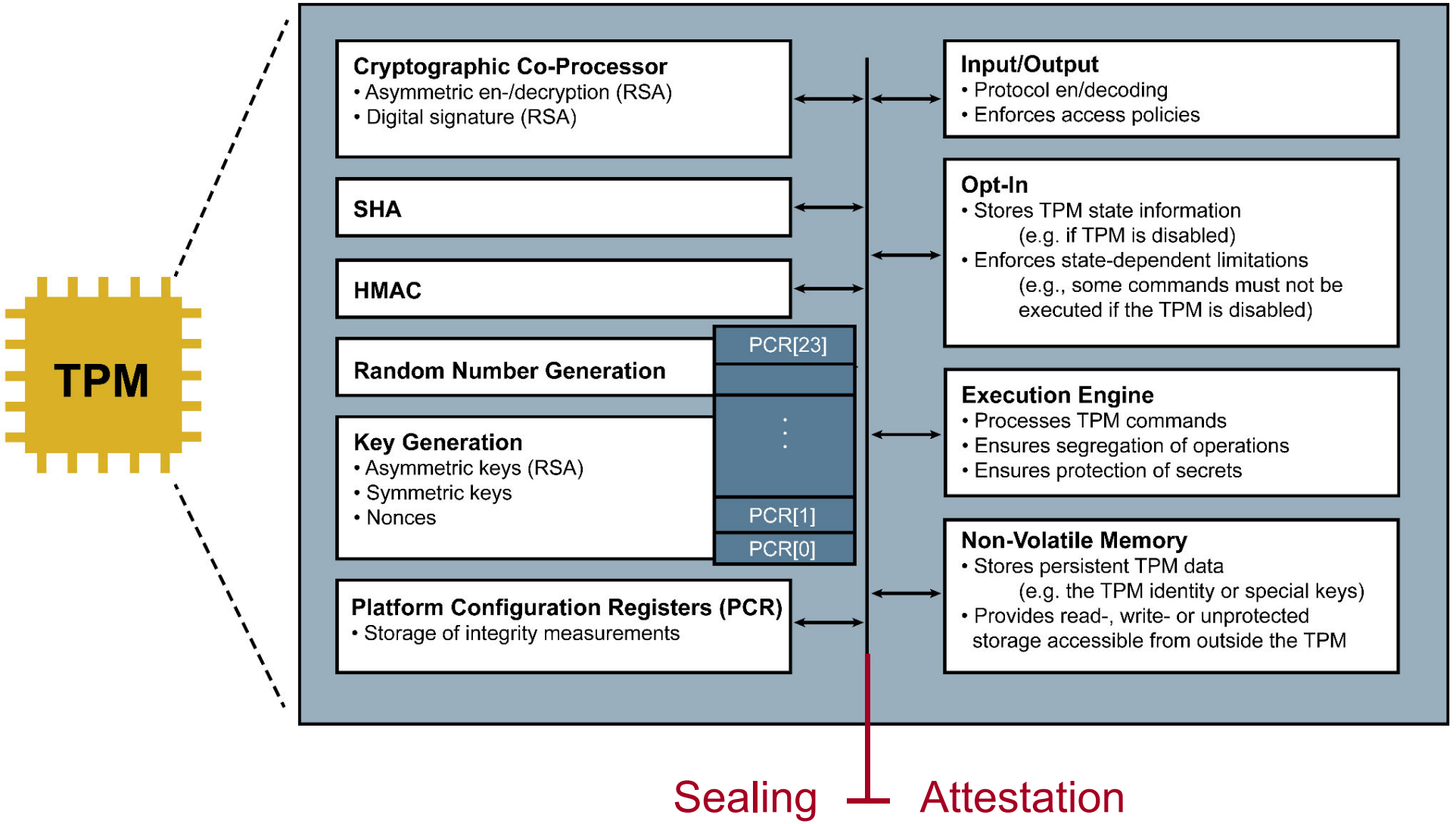
- **Nachteile** eines Mikrokernels:
  - Weniger Leistung durch mehr Kommunikation zwischen den Prozessen

Sicherheitsarchitektur
→ Kernelarchitekturen (3/3)

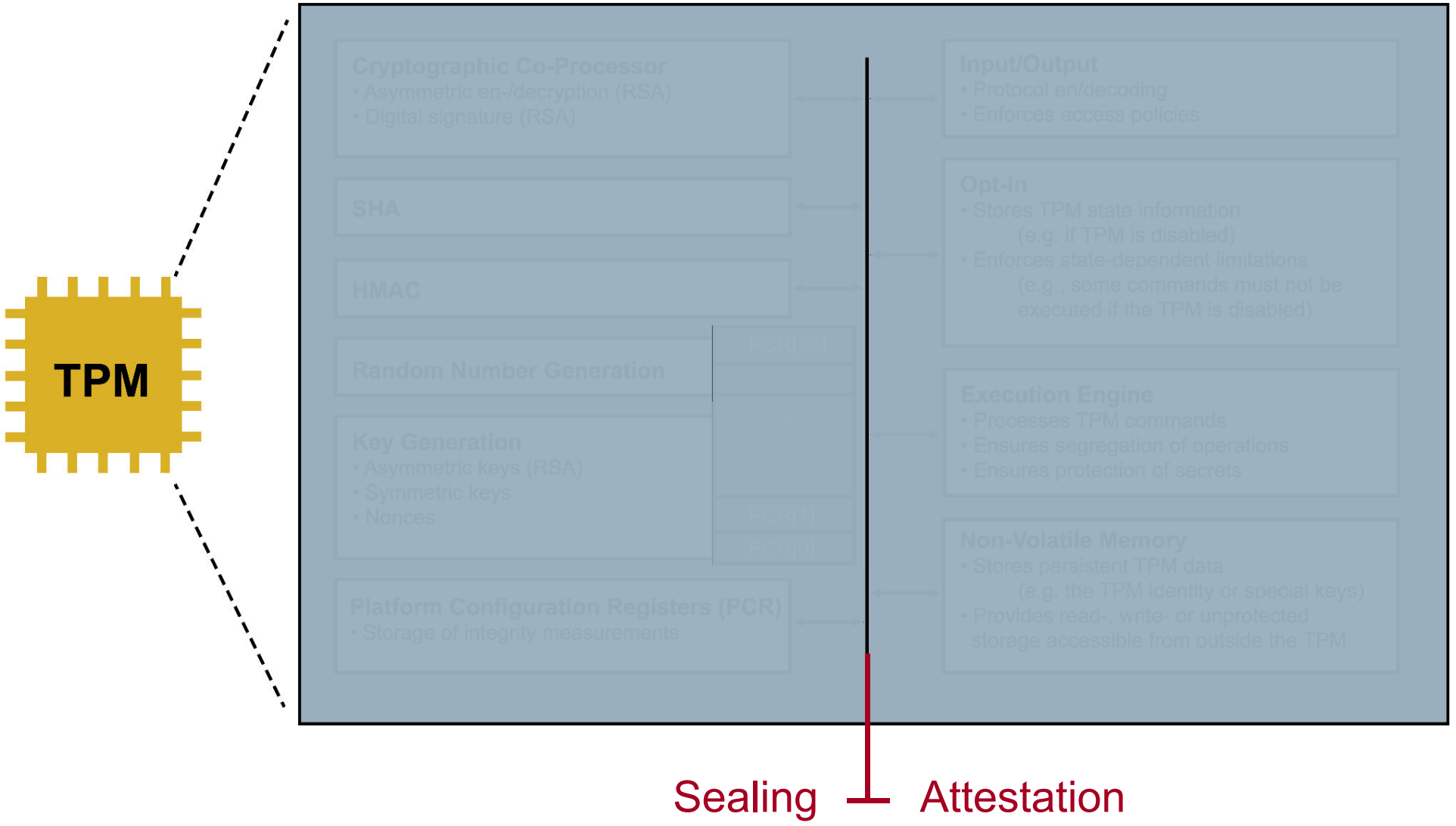
- **Vorteile** eines Mikrokernels:
  - Höhere Robustheit
  - Höhere Modularität
  - Höhere Flexibilität
  - Höhere IT-Sicherheit (kontrollierbare Interprozesskommunikation)
  - Weniger benötigter Speicherplatz

- **Nachteile** eines Mikrokernels:
  - Weniger Leistung durch mehr Kommunikation zwischen den Prozessen

HSM: Trusted Platform Module
→ Hardware-Sicherheitsanker



HSM: Trusted Platform Module
→ Hardware-Sicherheitsanker



Sicherheitsarchitektur

→ Core Root of Trust for Measurement (CRTM)

- Messvorgang über einzelne **Systemzustände** (Hard- und Software).
  - Speicherung der Messungen in den **PCRs**.

- **Authenticated Boot:**
  - Systemzustände messen.
  - Speicherung in den PCRs.
  - Überprüfung der Integrität.

- **Secure Boot:**
  - Systemzustände messen.
  - Überprüfung der Integrität.
  - Ggf. Bootvorgang stoppen.

"Transitives Vertrauen"

Sicherheitsarchitektur
→ Identitäten (1/2)

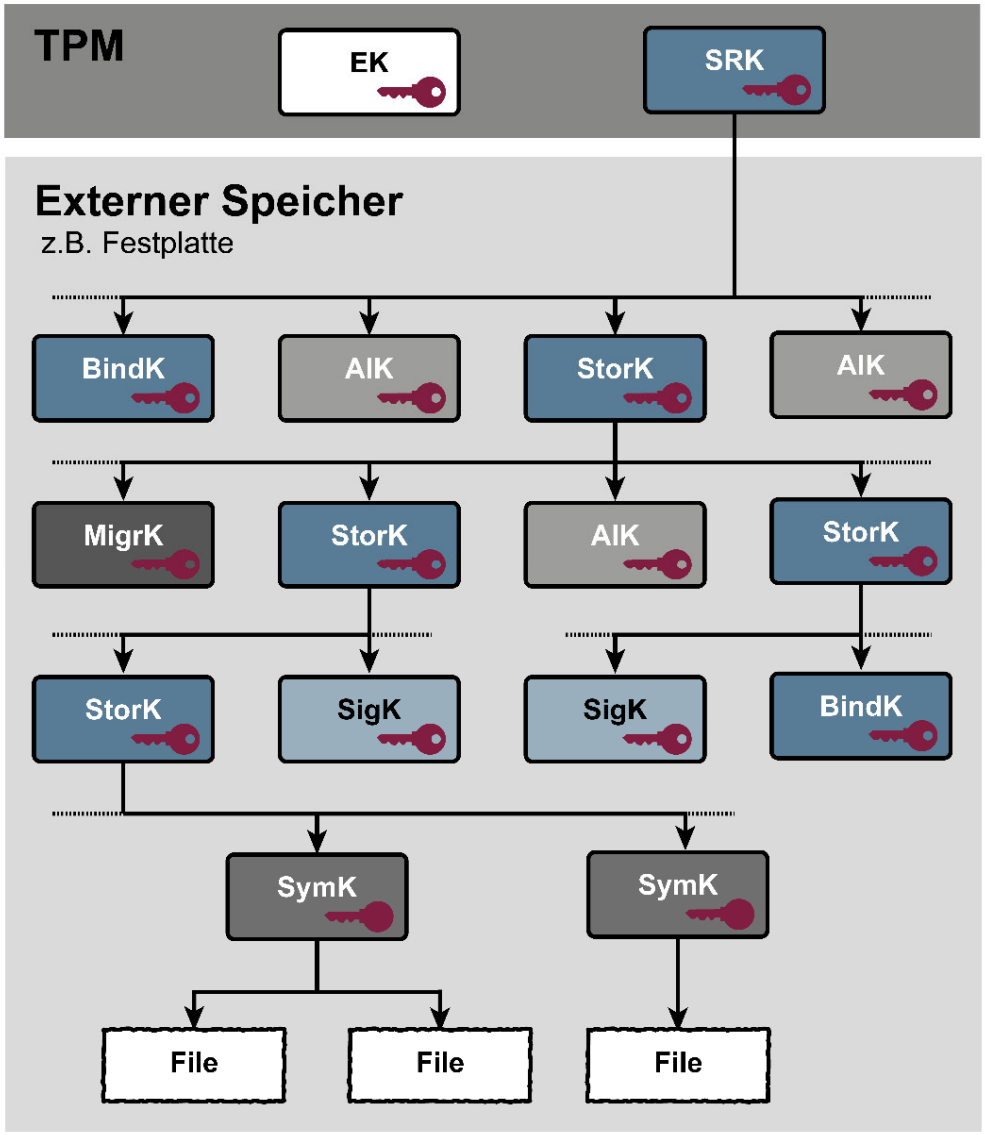
- **Endorsement Key (EK):**
  - Eindeutige TPM-Identität (nicht migrierbar).
  - RSA-Schlüsselpaar (im Herstellungsprozess erzeugt).
  - Geheimer Schlüssel im TPM gespeichert.
  - Öffentlicher Schlüssel ist datenschutzsensitiv.
  - TPM-Hersteller verwaltet PKI.

- **Endorsement Credential (EC):**
  - Elektronisches Zertifikat vom TPM-Hersteller.
  - Bestätigt ordnungsgemäße Erstellung und Einbettung des EK.
  - Bestandteile: TPM-Herstellername, TPM-Modellnummer, TPM-Version, Öffentlicher Schlüssel des EK.

Sicherheitsarchitektur
→ Identitäten (2/2)

- **Platform Identität (PI):**
  - Entspricht der TPM-Identität (EK).
  - Physikalische oder logische Bindung des TPMs an die Plattform (z.B. mittels anlöten an das Motherboard oder Kryptographie).
  - Plattform ≙ Motherboard/IT-System.
  - Plattform muss konform zur Evaluierungsrichtlinien der TCG sein → Conformance Credential (CC)
- **Platform Credential:**
  - Elektronisches Zertifikat vom Plattform-Hersteller.
  - Bestätigt gültige Verbindung zwischen TPM und Plattform → Trusted Plattform.
  - Bestandteile: Name des Plattformherstellers, Plattformmodell und Versionsnummer, Verweise auf die EC und CC.

Sicherheitsarchitektur
→ Schlüssel und deren Eigenschaften (1)



Sicherheitsarchitektur
→ Schlüssel und deren Eigenschaften (1)

Sicherheitsarchitektur
→ Schlüssel und deren Eigenschaften (2)

- **Migratable Keys** → Auf andere Plattformen übertragbar.
- **Non-Migratable Keys** → An die Plattform gebunden.

- **Storage Root Key (SRK):**
  - Wurzel der Schlüsselhierarchie.
  - Während der Installation des TPM-Eigentümers generiert.
  - Löschung des TPM-Eigentümers → Löschung des SRK → Kein Zugriff auf die Schlüsselhierarchie mehr.
  - **Eigenschaften**:
    - Steht im nicht flüchtigen Speicher des TPMs.
    - Ist nicht migrierbar.

Sicherheitsarchitektur
→ Schlüssel und deren Eigenschaften (3)

- **Attestation Identity Keys (AIK):**
  - Verwendet für die Trusted Computing Funktion „**Attestation**“:
    - Authentische Bestätigung der Integrität einer Plattformkonfiguration (z.B. aktuelle Hard- und Softwareumgebung).
  - Nötig, da EK datenschutzsensibel ist.
  - AIKs werden vom TPM-Besitzer generiert.
  - TPM/Plattform kann mehrere AIKs besitzen (z.B. für Online-Banking, E-Mail, …)
  - **Eigenschaften:**
    - Stehen im nicht flüchtigen Speicher des TPMs.
    - Nicht migrierbar.

Sicherheitsarchitektur
→ Schlüssel und deren Eigenschaften (4)

- **Storage Keys (StorK):**
  - Verschlüsselung von weiteren Schlüsseln und Daten außerhalb des TPMs.
  - Verwendet für die Trusted Computing Funktion „**Sealing**“:
    - Zustand der Plattform wird Teil der Verschlüsselung.
    - Entschlüsselung nur im vorher definierten Zustand möglich.
  - **Eigenschaften:**
    - RSA-Schlüsselpaar.
    - Im Allgemeinen darf die Migration zu anderen TPMs erfolgen.

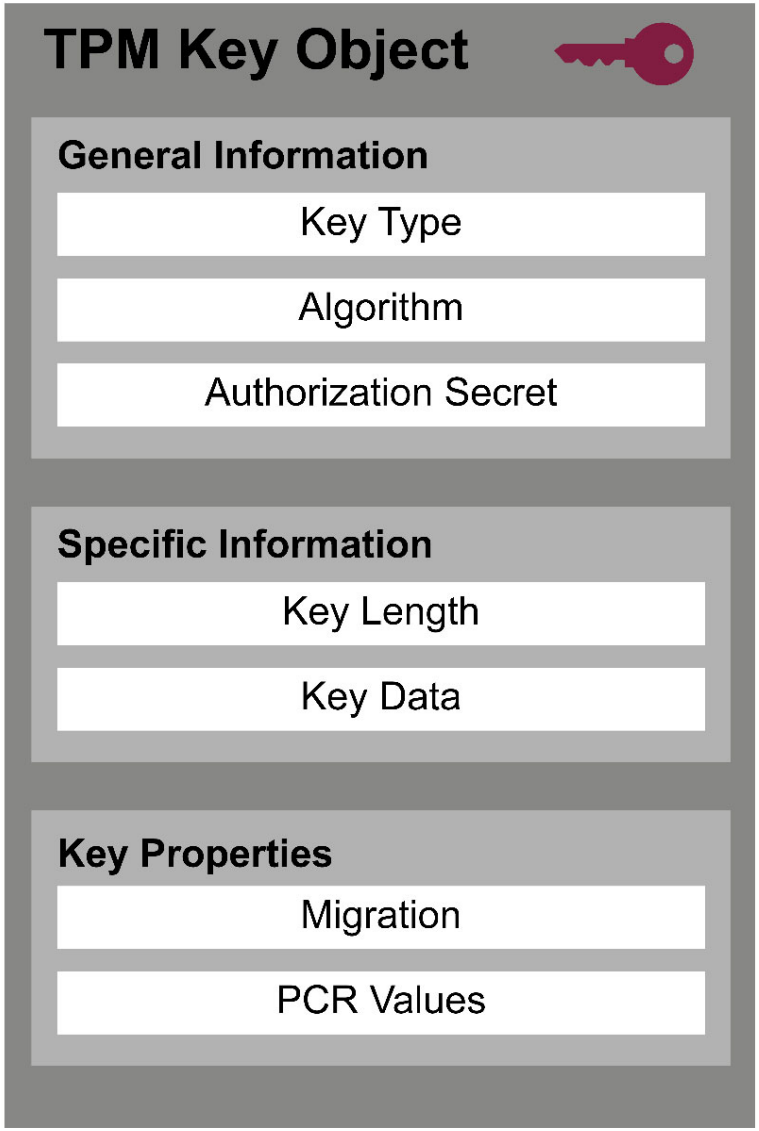
Sicherheitsarchitektur
→ Schlüssel und deren Eigenschaften (5)

- **Binding Keys (BindK):**
  - Verschlüsselung von beliebigen Daten außerhalb des TPMs.
  - Entspricht der asymmetrischen Verschlüsselung.
  - **Eigenschaften:**
    - RSA-Schlüsselpaar (es können auch andere Algorithmen vom TPM unterstützt werden).
    - Im Allgemeinen darf die Migration zu anderen TPMs erfolgen.
    - Kann nur mit Binding-Befehlen verwendet werden.

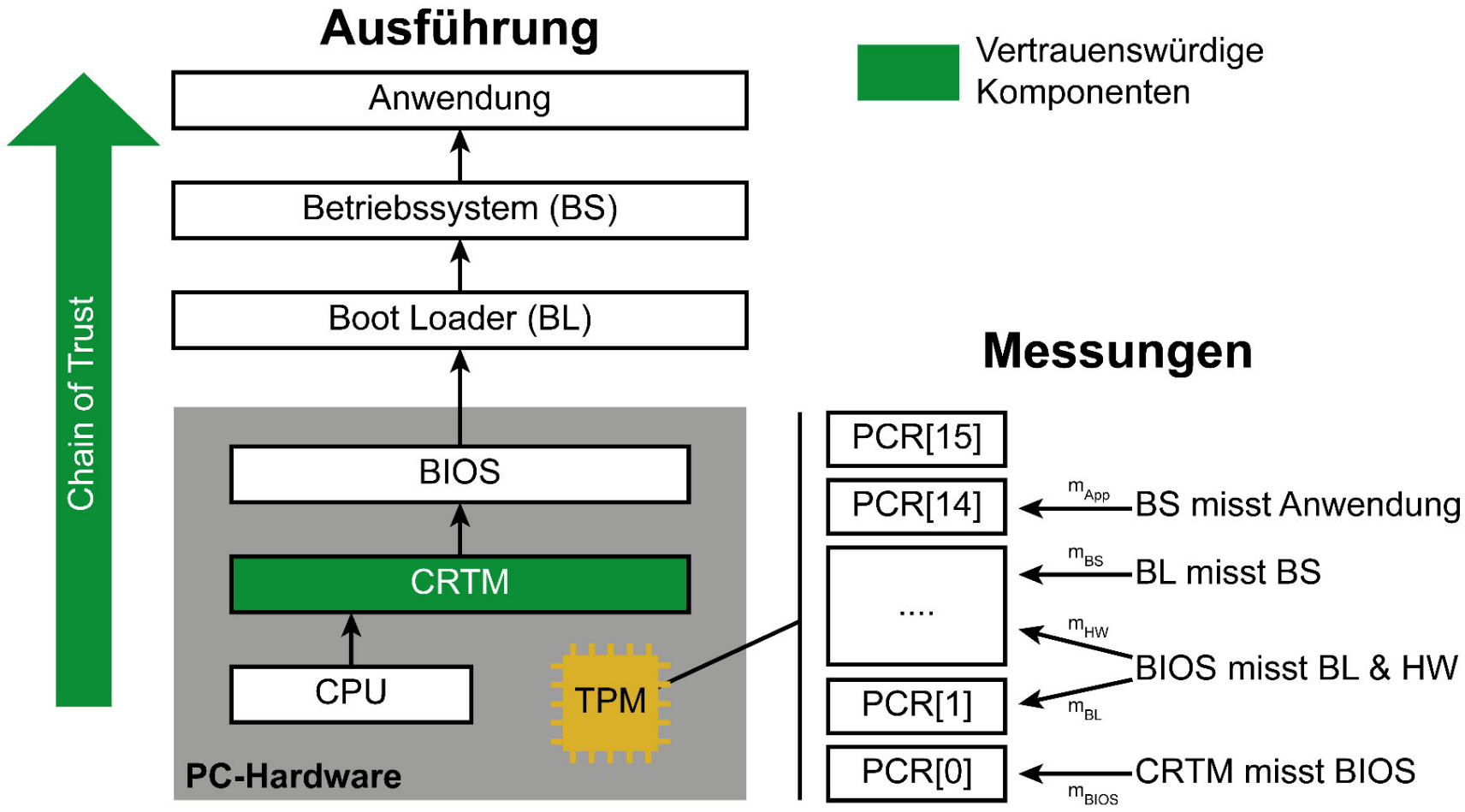
Sicherheitsarchitektur
→ Schlüssel und deren Eigenschaften (6)

- **Signing Keys (SigK):**
  - Nachweis der Authentizität und Integrität von beliebigen Daten /Protokollnachrichten innerhalb und außerhalb des TPMs.
  - **Eigenschaften:**
    - RSA-Schlüsselpaar (es können auch andere Algorithmen vom TPM unterstützt werden).
    - Im Allgemeinen darf die Migration zu anderen TPMs erfolgen.

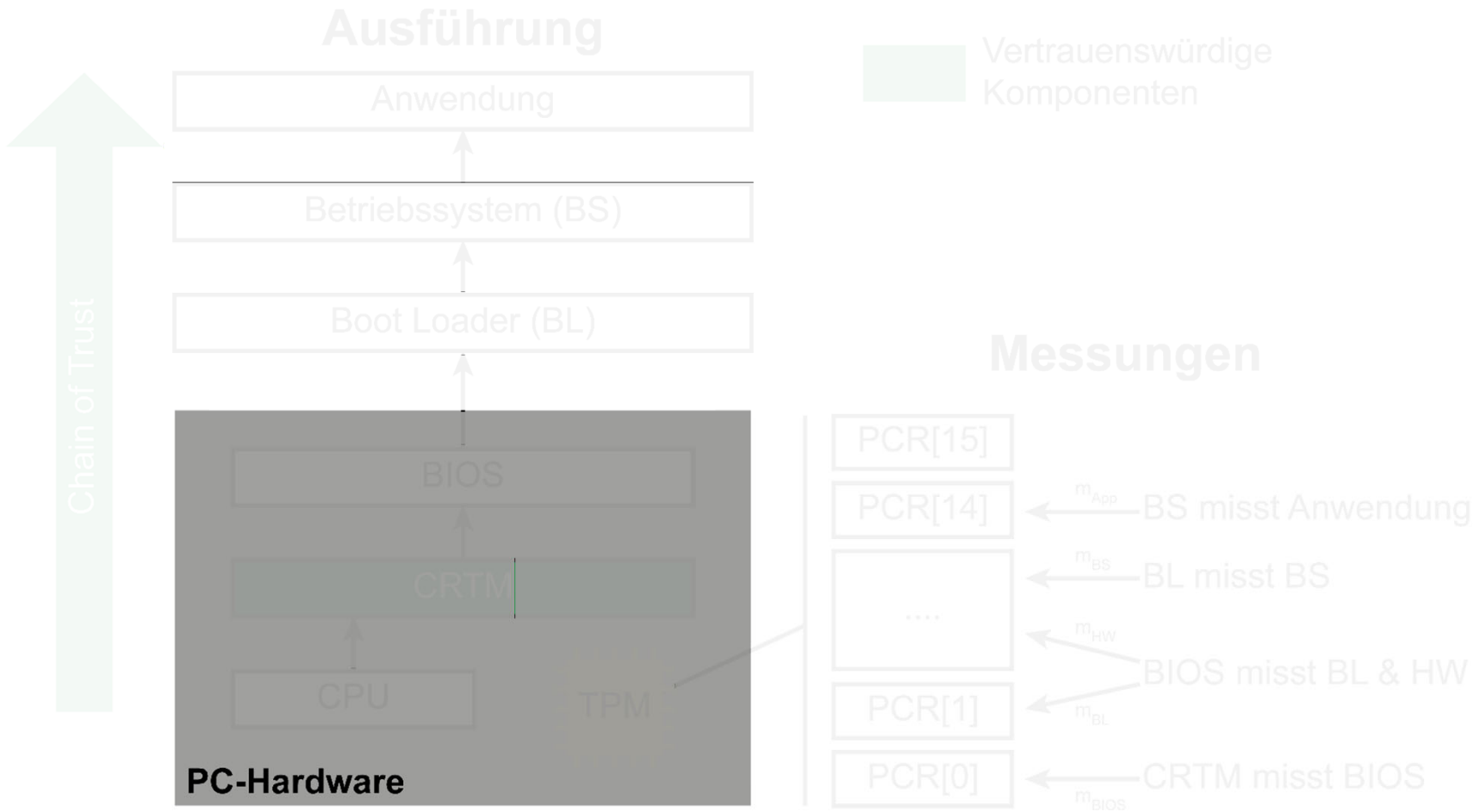
Sicherheitsarchitektur
→ Schlüssel und deren Eigenschaften (7)



Sicherheitsarchitektur
→ Authenticated Boot



Sicherheitsarchitektur
→ Authenticated Boot



Sicherheitsarchitektur
→ Sealing Funktionen und Parameter

Eingabe Parameter

daten	*{unverschlüsselte Daten}*

Ausgabe Parameter

cipher	*{verschlüsselte Daten}*
cryptedKEY	*{verschlüsselter Schlüssel}*

TPM Interne Funktionen und Daten

encrypt (key, daten)	*{symmetrischer Verschlüsselungsalgorithmus „AES“}*
H (daten)	*{One-Way-Hashfunktion “SHA-256”}*
genKey()	*{Schlüsselerzeugung}*
SRK	*{Storage Root Key}*
PCRs	*{PCR-0, PCR-1, …} z.B. aktuell abgespeicherte PCR-Werte*

plainKEY = genKEY ()

cipher = encrypt (plainKEY, (daten //H (daten //PCR-0 //… //PCR-x))

cryptedKEY = encrypt (SRK, plainKEY //H (plainKEY))

Sicherheitsarchitektur
→ Sealing Funktionen und Parameter

Eingabe Parameter

daten	*{unverschlüsselte Daten}*

Ausgabe Parameter

cipher	*{verschlüsselte Daten}*
cryptedKEY	*{verschlüsselter Schlüssel}*

TPM Interne Funktionen und Daten

encrypt (key, daten)	*{symmetrischer Verschlüsselungsalgorithmus „AES“}*
H (daten)	*{One-Way-Hashfunktion “SHA-256”}*
genKey()	*{Schlüsselerzeugung}*
SRK	*{Storage Root Key}*
PCRs	*{PCR-0, PCR-1, ···} z.B. aktuell abgespeicherte PCR-Werte*

plainKEY = genKEY ()

cipher = encrypt (plainKEY, (daten //H (daten //PCR-0 //··· //PCR-x))

cryptedKEY = encrypt (SRK, plainKEY //H (plainKEY))

Sicherheitsarchitektur
→ Un-Sealing Funktionen und Parameter

Eingabe Parameter

cipher	*{verschlüsselte Daten}*
cryptedKEY	*{verschlüsselter Schlüssel}*

Ausgabe Parameter

daten	*{unverschlüsselte Daten}*

TPM Interne Funktionen und Daten

decrypt (key, daten)	*{symmetrischer Verschlüsselungsalgorithmus „AES“}*
H (daten)	*{One-Way-Hashfunktion “SHA-256”}*
checkPCRs (Hash-Value)	*{vergleicht PCRs-Inhalte mit Hash-Value}*
SRK	*{Storage Root Key}*
PCRs	*{PCR-0, PCR-1, …}*

plainKEY = decrypt (SRK, cryptedKEY)

daten //H (daten //PCR-0 //… //PCR-x) = decrypt (plainKEY, cipher)

```
if ( checkPCRs ( Hash-Value ) )
        return daten
else
        return ERROR
```

Sicherheitsarchitektur
→ Un-Sealing Funktionen und Parameter

Eingabe Parameter

cipher	*{verschlüsselte Daten}*
cryptedKEY	*{verschlüsselter Schlüssel}*

Ausgabe Parameter

daten	*{unverschlüsselte Daten}*

TPM Interne Funktionen und Daten

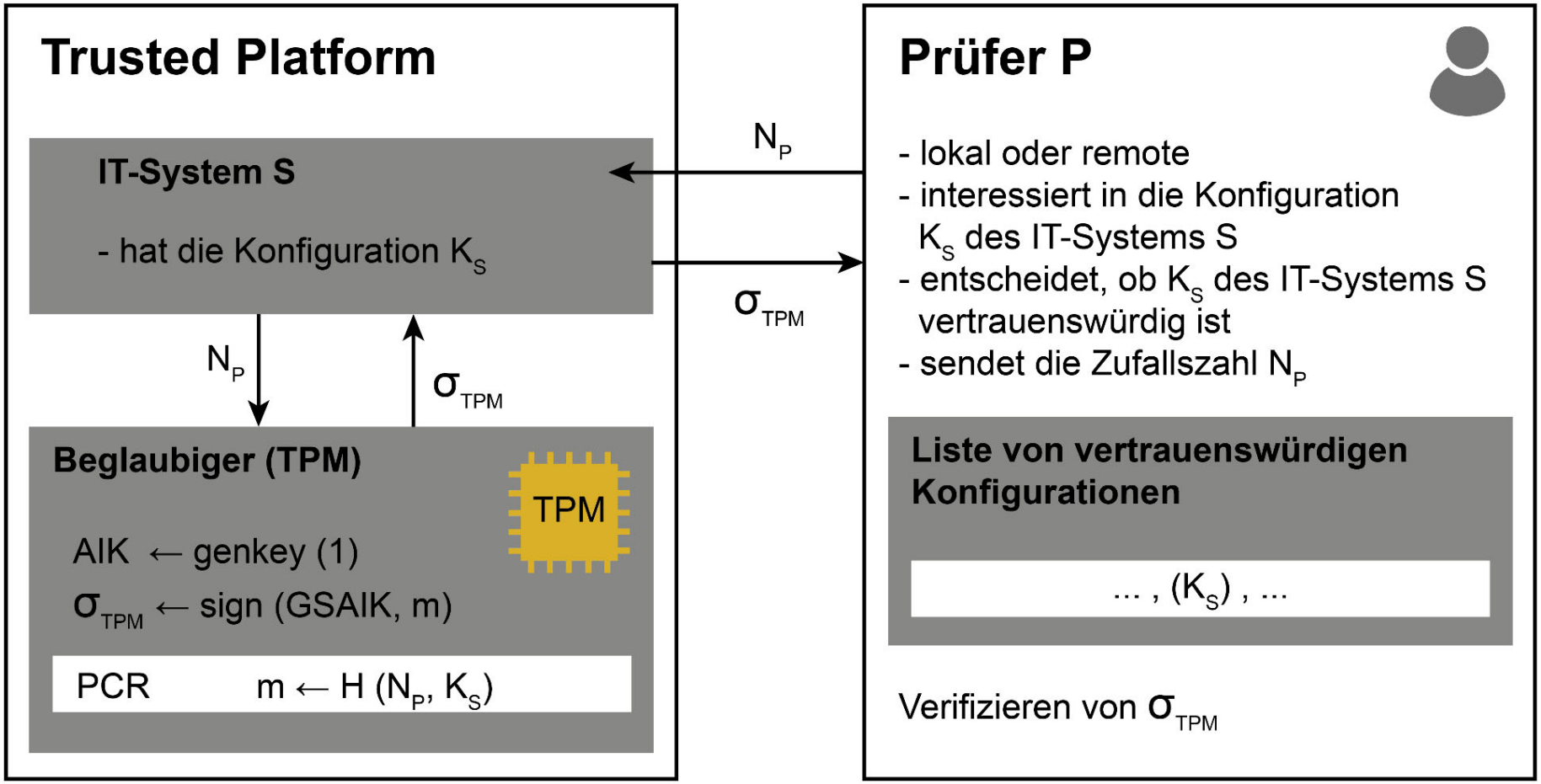
decrypt (key, daten)	*{symmetrischer Verschlüsselungsalgorithmus „AES“}*
H (daten)	*{One-Way-Hashfunktion “SHA-256”}*
checkPCRs (Hash-Value)	*{vergleicht PCRs-Inhalte mit Hash-Value}*
SRK	*{Storage Root Key}*
PCRs	*{PCR-0, PCR-1, ···}*

plainKEY = decrypt (SRK, cryptedKEY)

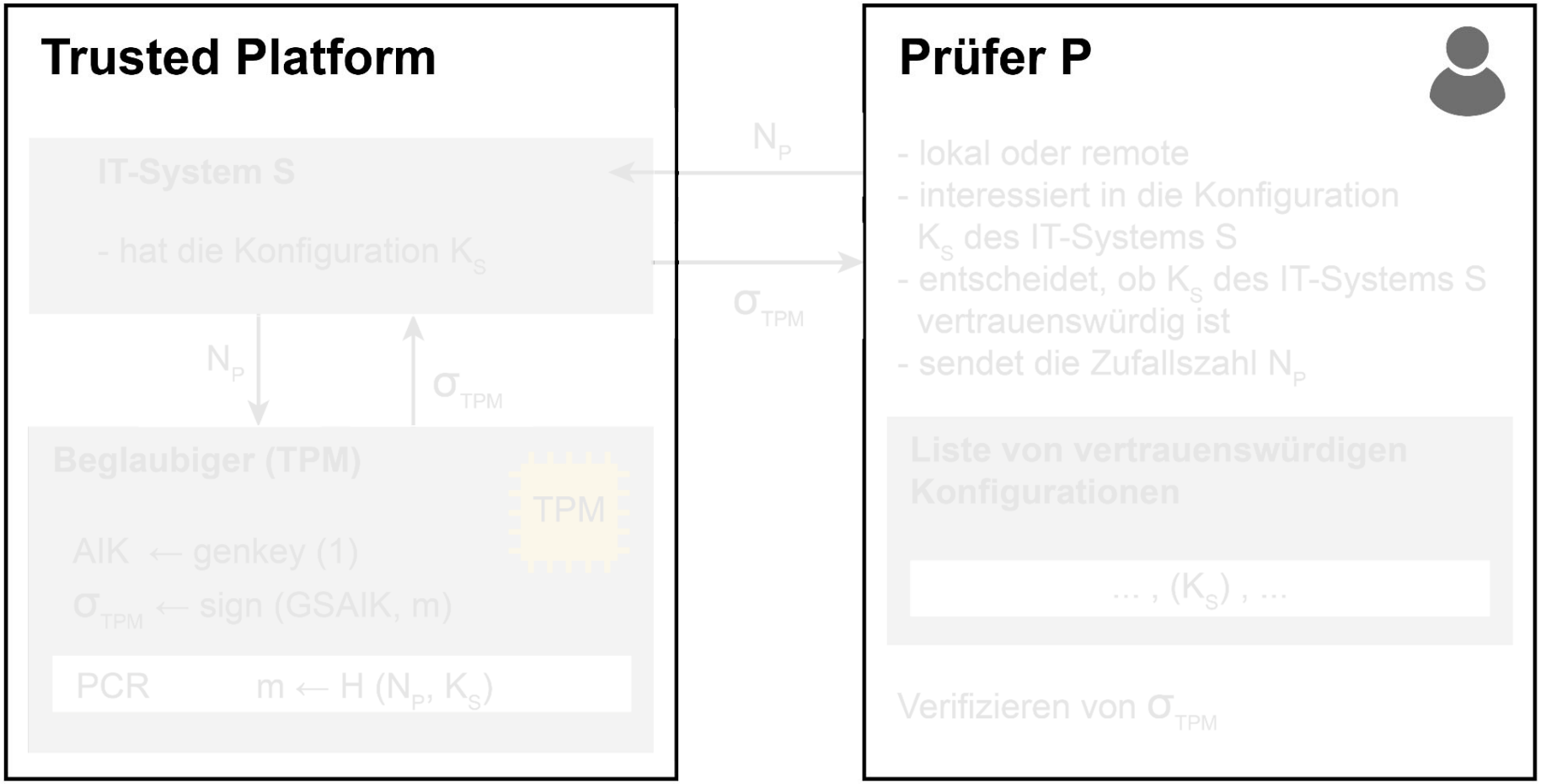
daten //H (daten //PCR-0 //··· //PCR-x) = decrypt (plainKEY, cipher)

```
if ( checkPCRs ( Hash-Value ) )
        return daten
else
        return ERROR
```

Sicherheitsarchitektur
→ (Remote) Attestation Ablauf



Sicherheitsarchitektur
→ (Remote) Attestation Ablauf



Sicherheitsarchitektur
→ Signaturfunktion für die Attestierung

Eingabe Parameter

random	*{Zufallszahl des Prüfers P – N_P}*

Ausgabe Parameter

signature//certificate	*{Signatur der aktuellen Systemkonfiguration des AIKs}*

TPM Interne Funktionen und Daten

sign (key, daten)	*{RSA-Signatur}*
H (daten)	*{One-Way-Hashfunktion "SHA-256"}*
GSAIK	*{geheimer AIK-RSA-Schlüssel}*
AIK-certificate	*{elektronisches Zertifikat des AIKs}*
PCRs	*{PCR-0, PCR-1, …} z.B. aktuell abgespeicherte PCR-Werte*

σ_{TPM} = sign (GSAIK, H (random //PCR-0 //… //PCR-x))

Sicherheitsarchitektur
→ Signaturfunktion für die Attestierung

Eingabe Parameter

random	*{Zufallszahl des Prüfers P – N_P}*

Ausgabe Parameter

signature//certificate	*{Signatur der aktuellen Systemkonfiguration des AIKs}*

TPM Interne Funktionen und Daten

sign (key, daten)	*{RSA-Signatur}*
H (daten)	*{One-Way-Hashfunktion "SHA-256"}*
GSAIK	*{geheimer AIK-RSA-Schlüssel}*
AIK-certificate	*{elektronisches Zertifikat des AIKs}*
PCRs	*{PCR-0, PCR-1, ···} z.B. aktuell abgespeicherte PCR-Werte*

σ_{TPM} = *sign (GSAIK, H (random //PCR-0 //··· //PCR-x))*

Sicherheitsarchitektur

→ Verifikationsfunktion für die Attestierung

Eingabe Parameter

signature//certificate	*{Signatur der Systemkonfiguration des AIKs}*

Ausgabe Parameter

return value	*{Rückgabewert}*

TPM Interne Funktionen und Daten

very (key, daten)	*{RSA-Signatur-Verifikation}*
H (daten)	*{One-Way-Hashfunktion "SHA-256"}*
ÖSAIK	*{öffentlicher AIK-RSA-Schlüssel}*
checkPCRs (PCR-Values)	*{vergleicht den Inhalt der PCRs mit den gewünschten Werten}*
PCRs	*{PCR-0, PCR-1, …}*

if (very (ÖSAIK, σ_{TPM})) and
if (checkCERT (certificate)) and
if (checkPCRs (Hash-Value))
return ok
else
return ERROR

Sicherheitsarchitektur

→ Verifikationsfunktion für die Attestierung

Eingabe Parameter

signature//certificate	*{Signatur der Systemkonfiguration des AIKs}*

Ausgabe Parameter

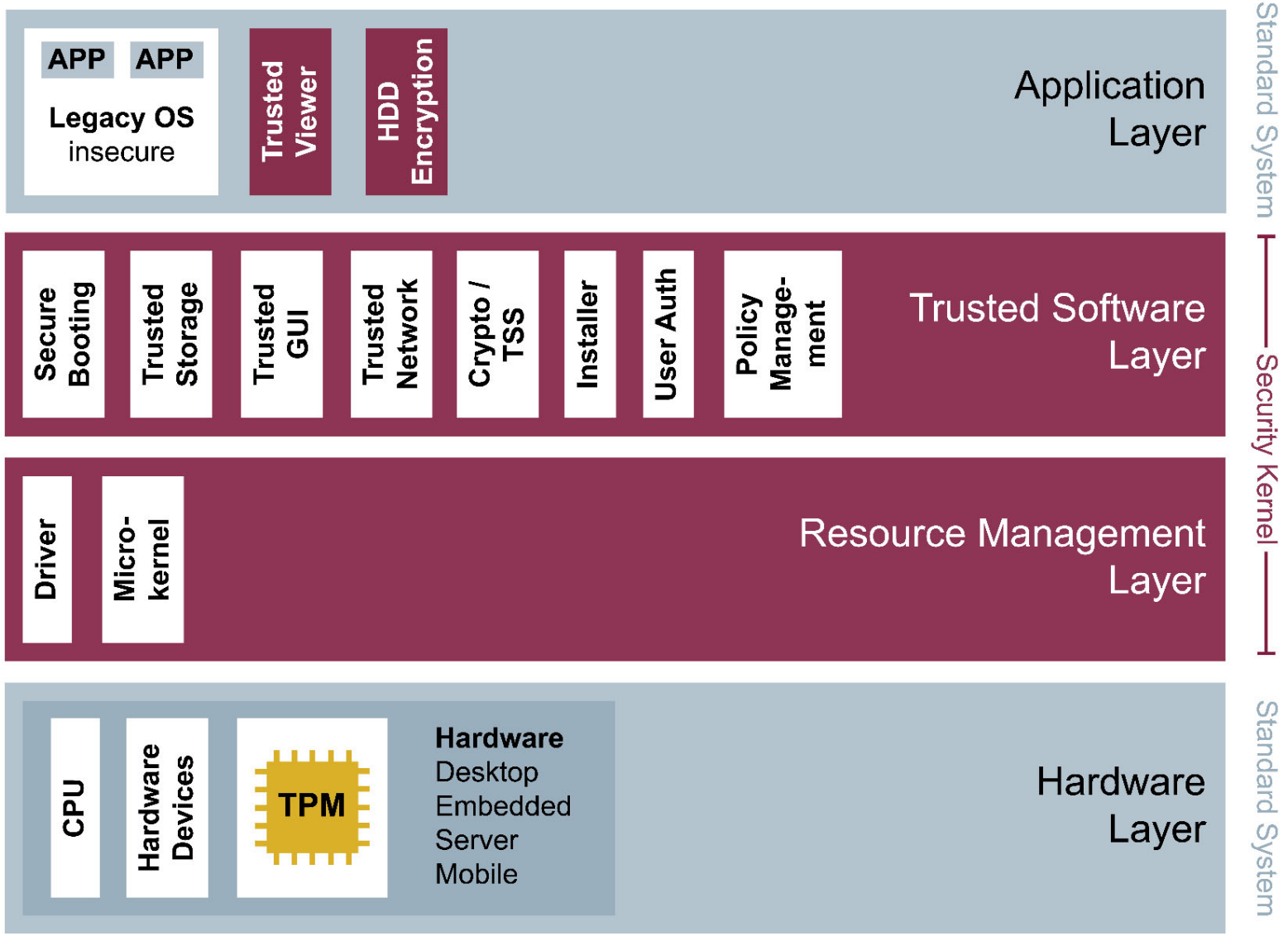
return value	*{Rückgabewert}*

TPM Interne Funktionen und Daten

very (key, daten)	*{RSA-Signatur-Verifikation}*
H (daten)	*{One-Way-Hashfunktion "SHA-256"}*
ÖSAIK	*{öffentlicher AIK-RSA-Schlüssel}*
checkPCRs (PCR-Values)	*{vergleicht den Inhalt der PCRs mit den gewünschten Werten}*
PCRs	*{PCR-0, PCR-1, ···}*

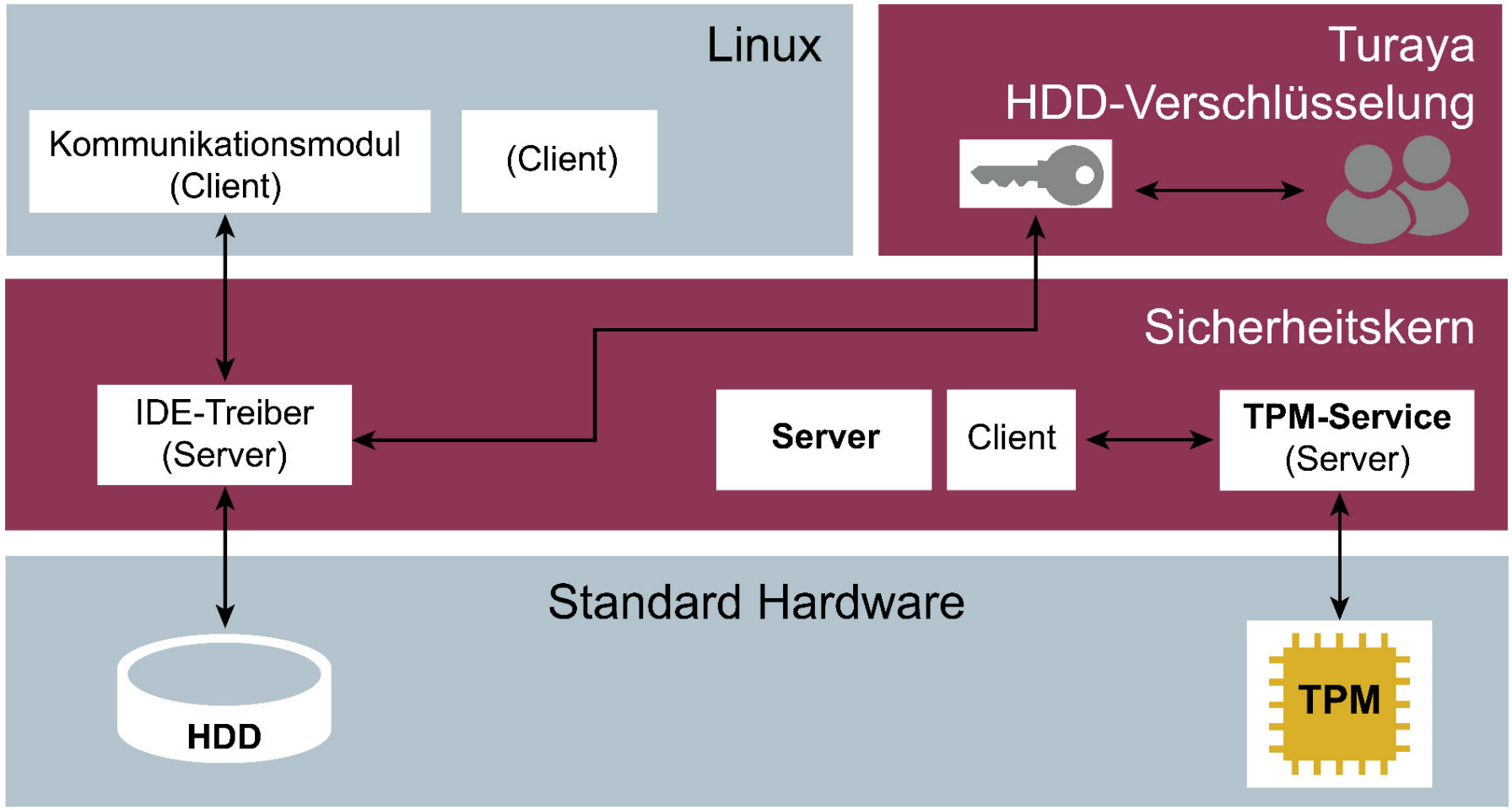
if (very (ÖSAIK, σ_{TPM})) and
if (checkCERT (certificate)) and
if (checkPCRs (Hash-Value))
return ok
else
return ERROR

Sicherheitsarchitektur
→ Trusted Plattform



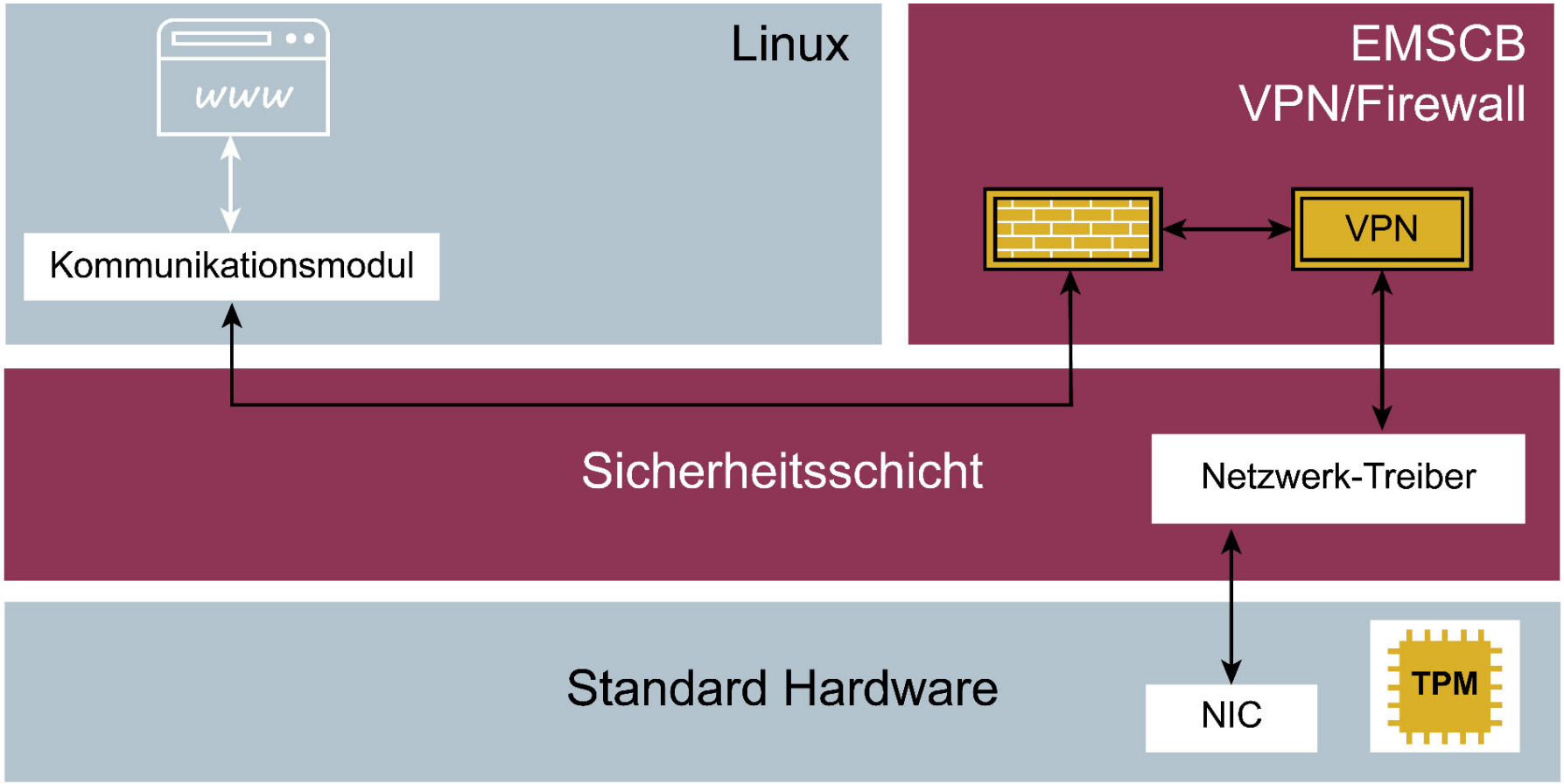
Sicherheitsarchitektur
→ Trusted Plattform

Sicherheitsarchitektur
→ Beispielanwendung: Turaya.Crypt



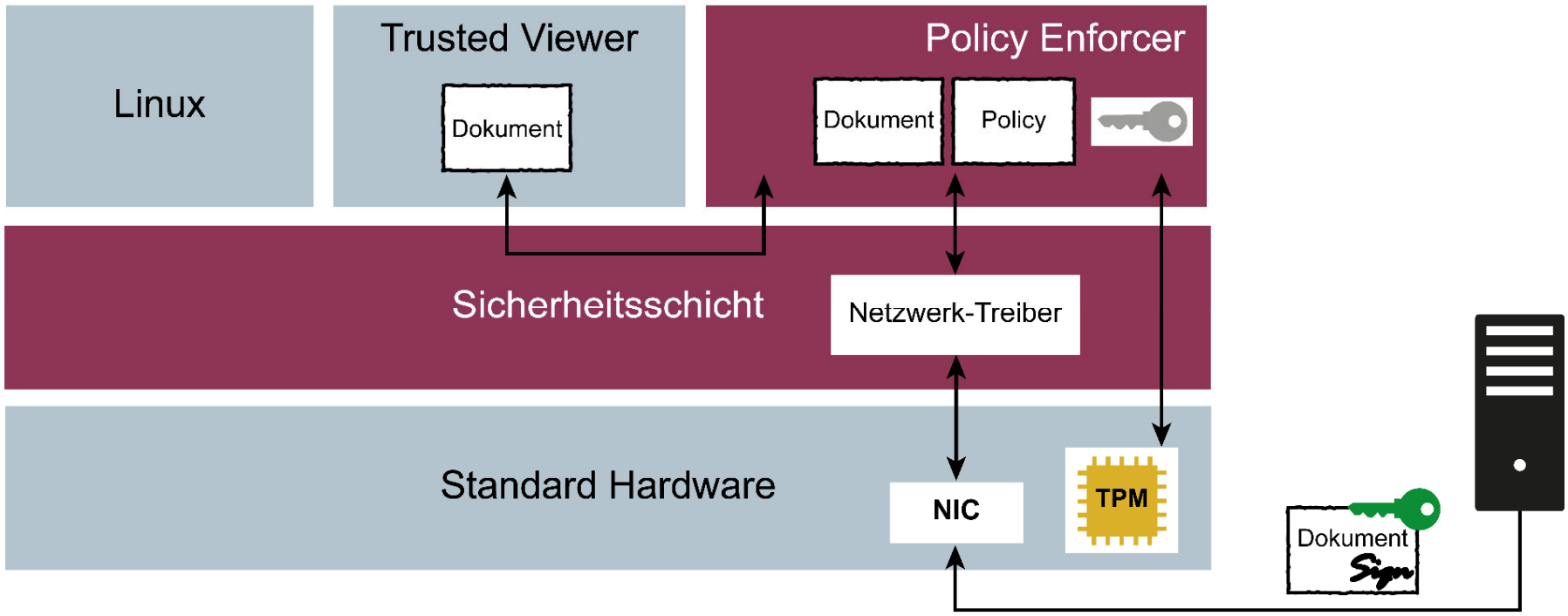
Sicherheitsarchitektur
→ Beispielanwendung: Turaya.Crypt

Sicherheitsarchitektur
→ Beispielanwendung: Turaya.VPN



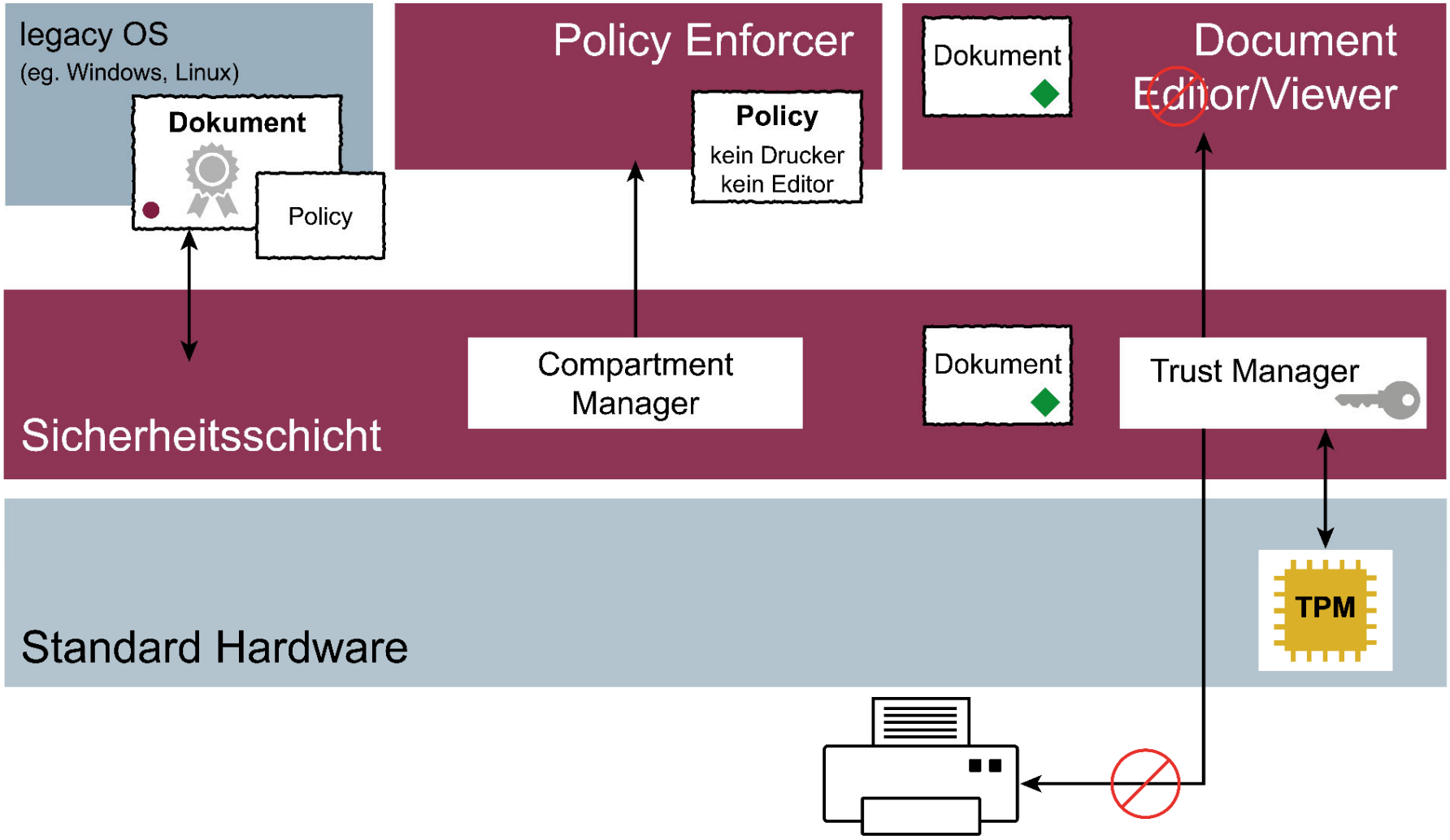
Sicherheitsarchitektur
→ Beispielanwendung: Turaya.VPN

Sicherheitsarchitektur
→ Beispielanwendung: Turaya.FairDRM

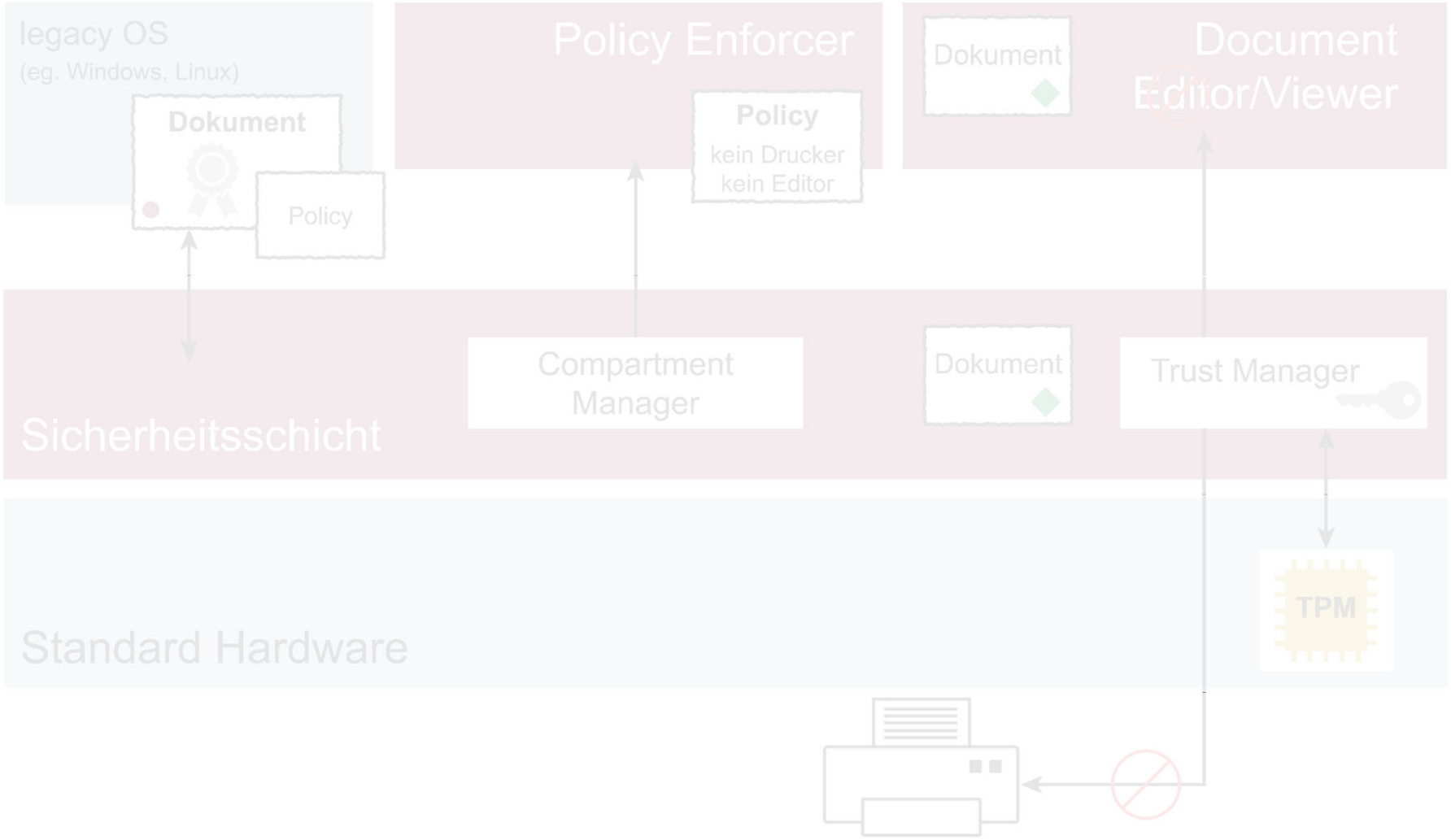


Sicherheitsarchitektur
→ Beispielanwendung: Turaya.FairDRM

Sicherheitsarchitektur
→ Beispielanwendung: Turaya.FairDRM



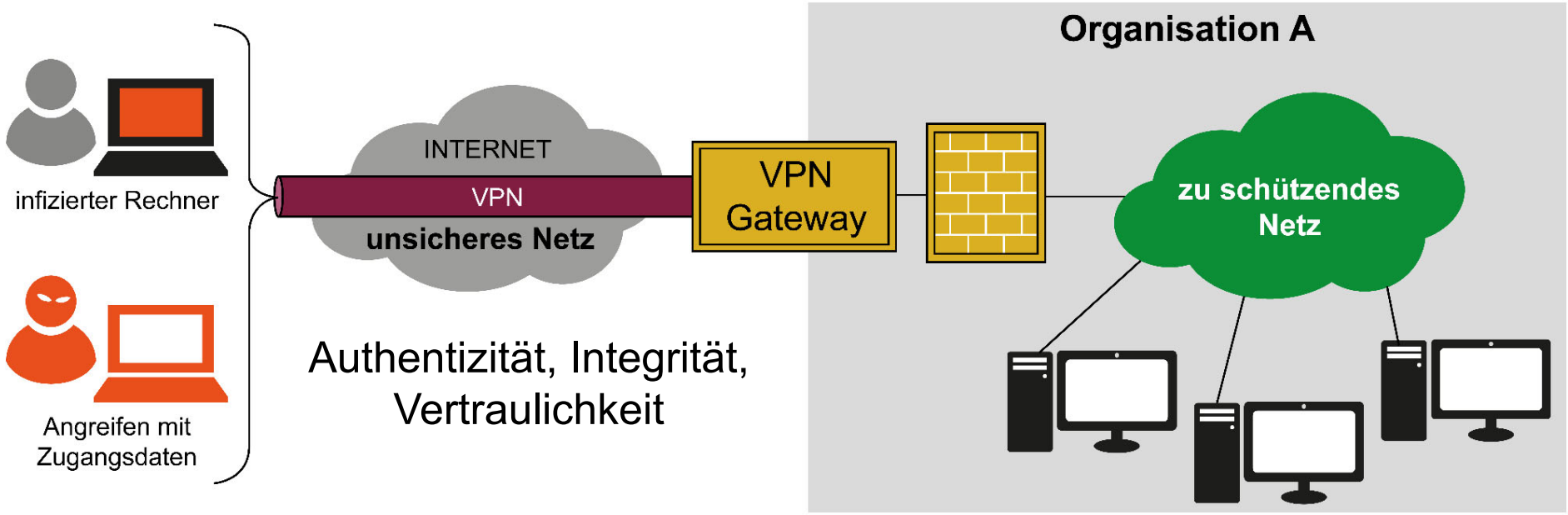
Sicherheitsarchitektur
→ Beispielanwendung: Turaya.FairDRM



Trusted Computing
→ Inhalt

- Ziele und Ergebnisse der Vorlesung
- Kernfunktionalitäten
- Sicherheitsarchitektur
- **Trusted Network Connect**
- Zusammenfassung

Trusted Network Connect
→ Problemstellung



Lösungsansätze:
Microsoft NAP,
Cisco NAC,
Trusted Computing Group (TCG) → **Trusted Network Connect (TNC)**

Trusted Network Connect
→ Problemstellung

Vertrauen?

Authentizität, Integrität, Vertraulichkeit

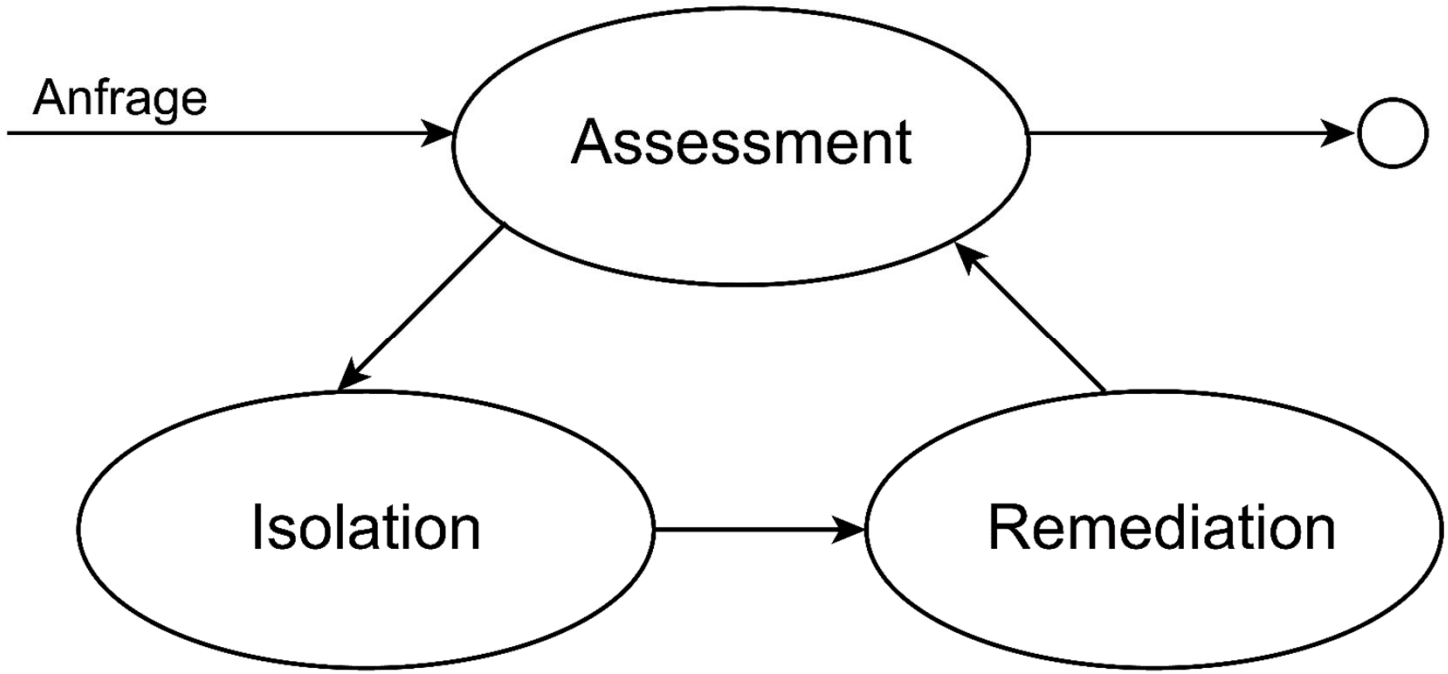
Lösungsansätze:
Microsoft NAP,
Cisco NAC,
Trusted Computing Group (TCG) → **Trusted Network Connect (TNC)**

Trusted Network Connect

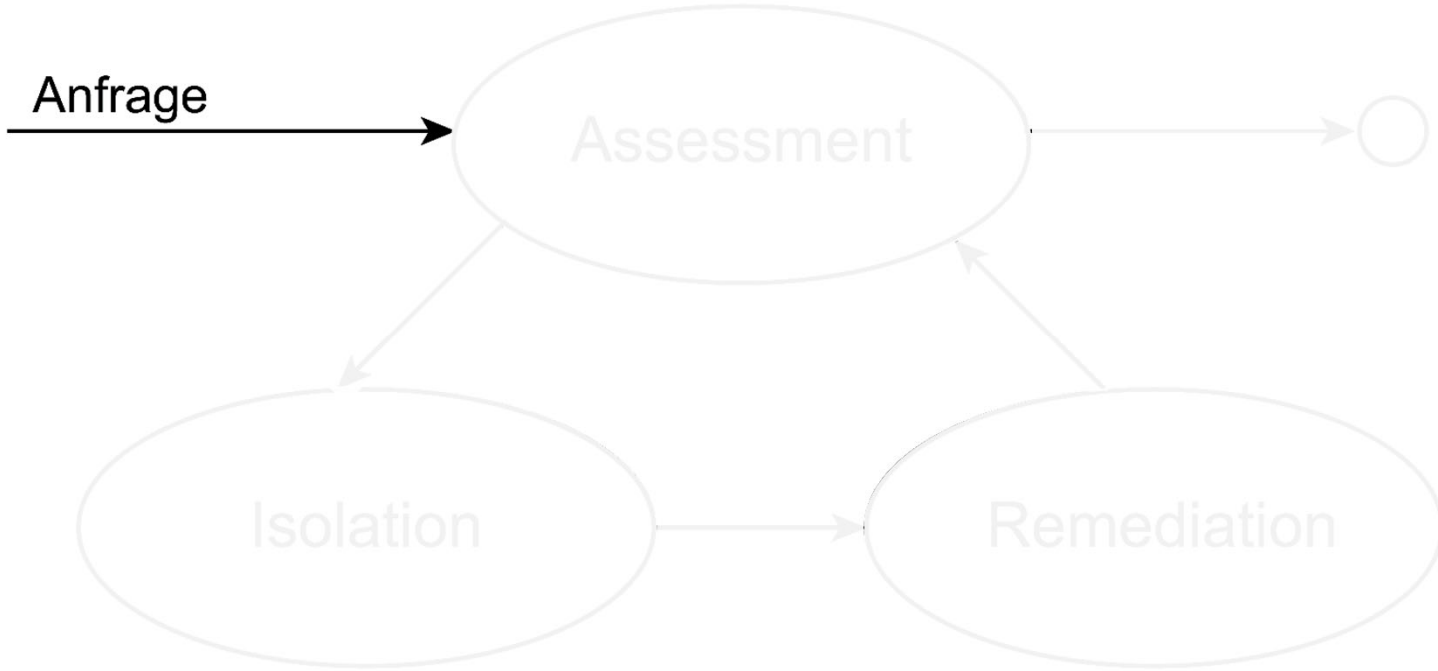
→ Vertrauenswürdige Netzwerkverbindungen

- Vertrauenswürdigkeit abhängig von der **Integrität**.
- **Alle** beteiligten **Kommunikationspartner** betrachten.
- **Kommunikationsrichtung** getrennt betrachten.
- Alle IT-Systeme, Infrastrukturelemente und das Umfeld der Kommunikation bewerten.
- Anforderungen in **Policies** definieren (z.B. erlaubte Konfigurationen, Betriebssysteme, Hard- und Software, …).
- Überprüfung **vor dem** Zugriff (**Prävention**).

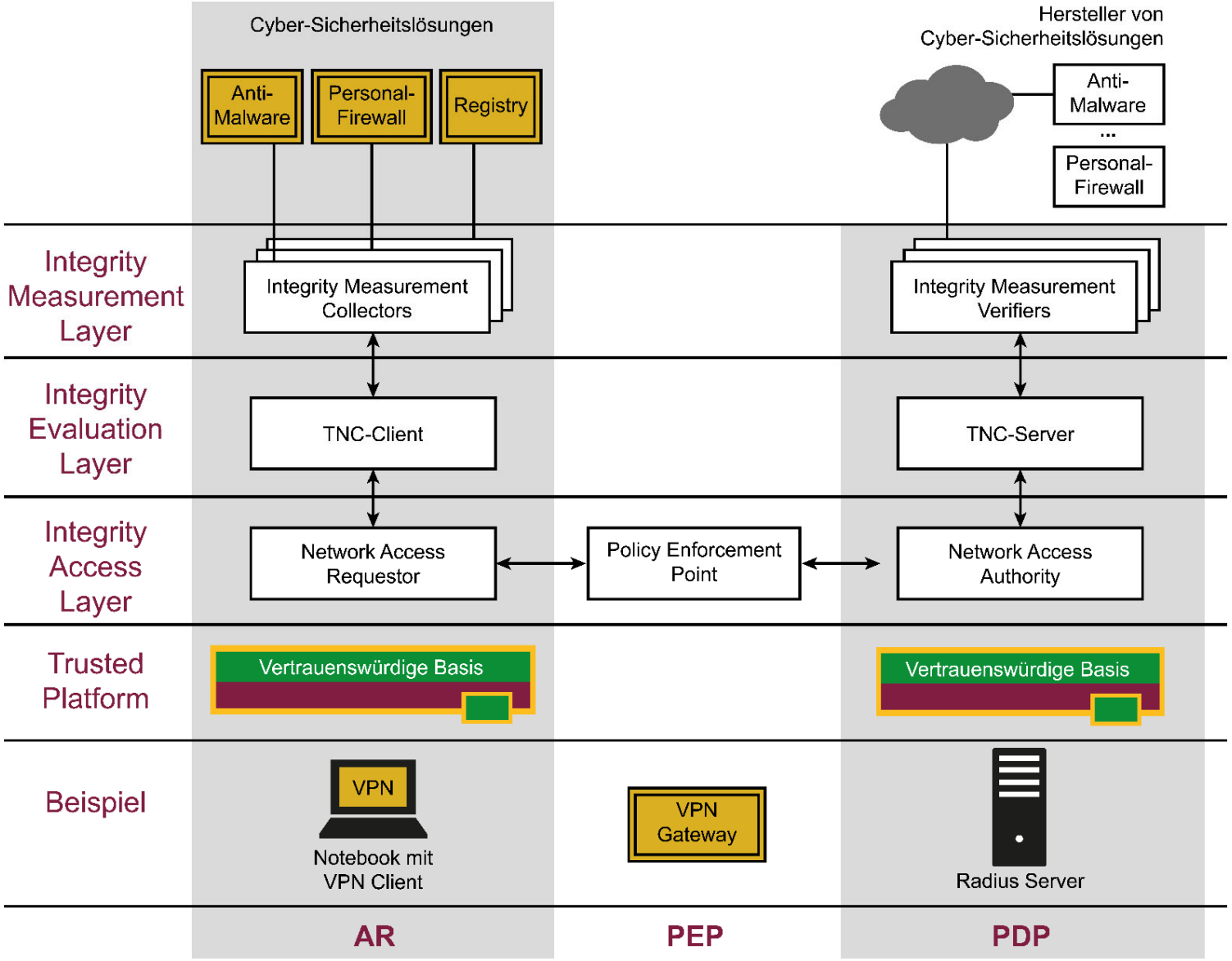
Trusted Network Connect
→ Phasen



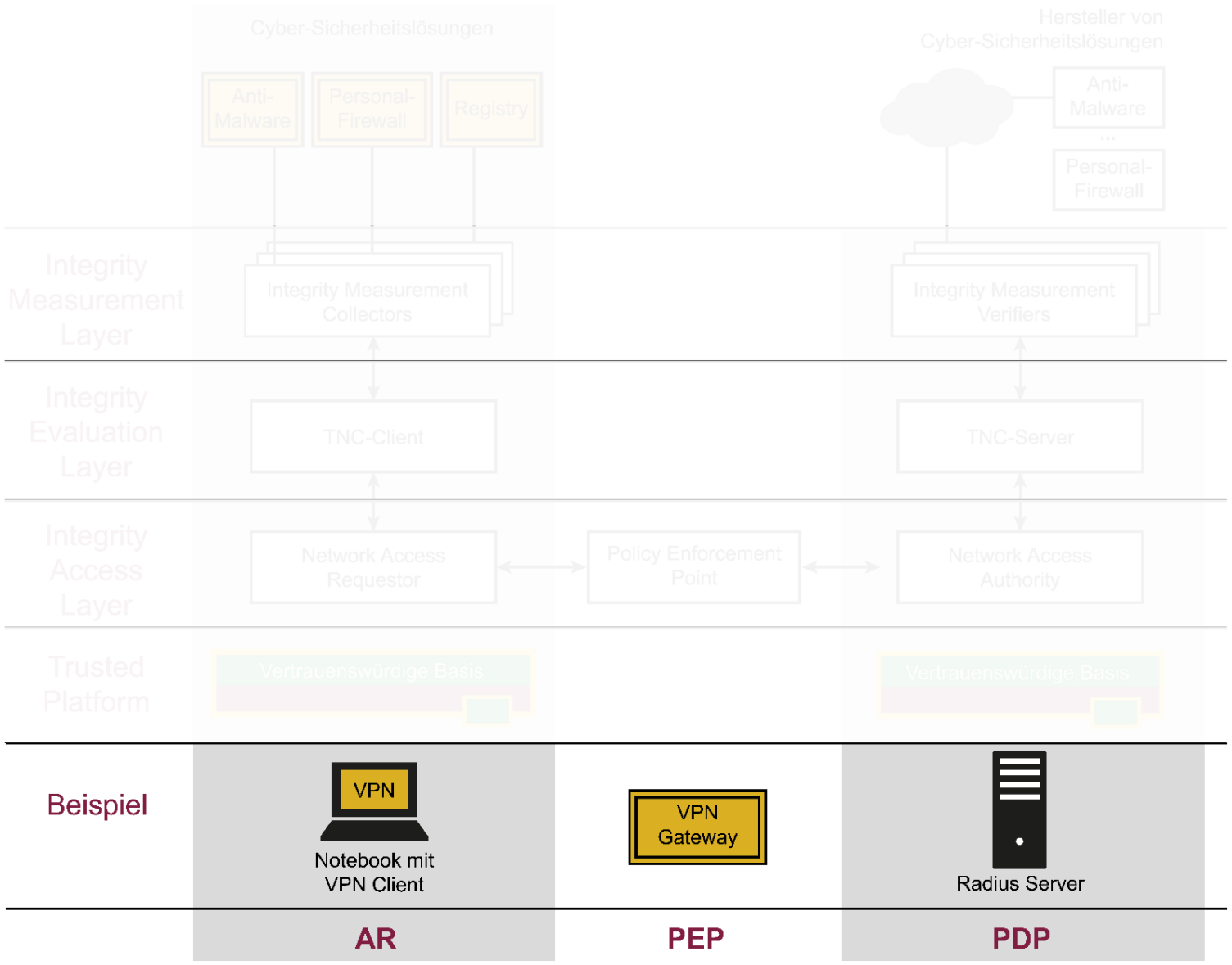
Trusted Network Connect
→ Phasen



Trusted Network Connect
→ Struktur



Trusted Network Connect
→ Struktur



Trusted Computing
→ Inhalt

- Ziele und Ergebnisse der Vorlesung
- Kernfunktionalitäten
- Sicherheitsarchitektur
- Trusted Network Connect
- **Zusammenfassung**

Trusted Computing
→ Zusammenfassung (1/2)

- Die Kernfunktionalitäten von Trusted Computing sind:
  - **Robustheit und Modularität**
  - **Integritätsüberprüfung**
  - **Trusted Process**
  - **Trusted Plattform**

- Vor- und Nachteile der verschiedenen **Kernelarchitekturen** müssen miteinander abgewogen werden.

- **CRTM** ist die Vertrauensbasis. Das Vertrauen ist **transitiv**.

- Die **TPM Schlüsselhierarchie** ermöglicht eine sichere Speicherung von Daten, auch auf externen Speichermedien.

Trusted Computing
→ Zusammenfassung (2/2)

- Wichtige Trusted Computing Funktionen sind:
  - **Authenticated Boot**
  - **Binding**
  - **Sealing**
  - **Attestation**
- Vertrauenswürdige Netzwerkverbindungen können durch **Trusted Network Connect (TNC)** realisiert werden.
- Die Festlegung einer sicheren und vertrauenswürdigen **Systemkonfiguration** ist mit zahlreichen Schwierigkeiten (technisch und politisch) verbunden.

Westfälische Hochschule
Gelsenkirchen Bocholt Recklinghausen
University of Applied Sciences

Trusted Computing

- Vorlesung Cyber-Sicherheit -

Prof. Dr. (TU NN)
Norbert Pohlmann

Institut für Internet-Sicherheit – if(is)
Westfälische Hochschule, Gelsenkirchen
http://www.internet-sicherheit.de

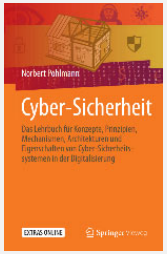
Anhang / Credits

Wir empfehlen

- **Kostenlose App securityNews**

securityNews

- **7. Sinn im Internet (Cyberschutzraum)**
  https://www.youtube.com/cyberschutzraum
- **Master Internet-Sicherheit**
  https://it-sicherheit.de/master-studieren/
- **Cyber-Sicherheit**
  Das **Lehrbuch** für Konzepte, Mechanismen, Architekturen und Eigenschaften von Cyber-Sicherheitssystemen in der Digitalisierung“, Springer Vieweg Verlag, Wiesbaden 2019
- https://norbert-pohlmann.com/cyber-sicherheit/



Besuchen und abonnieren Sie uns :-)

WWW
https://www.internet-sicherheit.de

Facebook
https://www.facebook.com/Internet.Sicherheit.ifis

Twitter
https://twitter.com/_ifis

YouTube
https://www.youtube.com/user/InternetSicherheitDE/

Prof. Norbert Pohlmann
https://norbert-pohlmann.com/

Quellen Bildmaterial

Eingebettete Piktogramme:
- Institut für Internet-Sicherheit – if(is)

Der Marktplatz IT-Sicherheit

(IT-Sicherheits-) Anbieter, Lösungen, Jobs, Veranstaltungen und Hilfestellungen (Ratgeber, IT-Sicherheitstipps, Glossar, u.v.m.) leicht & einfach finden.
https://www.it-sicherheit.de/

Literatur
→ Artikel / Bücher

N. Pohlmann, A.-R. Sadeghi, C. Stüble: „European Multilateral Secure Computing Base", DuD Datenschutz und Datensicherheit – Recht und Sicherheit in Informationsverarbeitung und Kommunikation, Vieweg Verlag, 09/2004
https://norbert-pohlmann.com/wp-content/uploads/2015/08/148-European-Multilateral-Secure-Computing-Base.pdf

M. Jungbauer, N. Pohlmann: „Vertrauenswürdige Netzwerkverbindungen mit Trusted Computing - Sicher vernetzt?“, IT-Sicherheit – Management und Praxis, DATAKONTEXT-Fachverlag, 06/2006
https://norbert-pohlmann.com/wp-content/uploads/2015/08/188-Vertrauenswürdige-Netzwerkverbindungen-mit-Trusted-Computing-–-Sicher-vernetzt-Prof.-Norbert-Pohlmann.pdf

M. Linnemann, N. Pohlmann: „Turaya - Die offene Trusted Computing Sicherheitsplattform", in "Open Source Jahrbuch 2007", Hrsg.: B. Lutterbeck, …, Lehmanns Media, Berlin, 2007
https://norbert-pohlmann.com/wp-content/uploads/2015/08/181-Turaya-–-An-Open-Trusted-Computing-Platform-Prof.-Pohlmann.pdf

M. Jungbauer, N. Pohlmann: „Integrity Check of Remote Computer Systems - Trusted Network Connect". In Proceedings of the ISSE/SECURE 2007 - Securing Electronic Business Processes - Highlights of the Information Security Solutions Europe/Secure 2007 Conference, Eds.: N. Pohlmann, H. Reimer, W. Schneider; Vieweg Verlag, Wiesbaden 2007
https://norbert-pohlmann.com/wp-content/uploads/2015/08/190-Integrity-Check-of-Remote-Computer-Systems-–-Trusted-Network-Connect-Prof.-Norbert-Pohlmann.pdf

N. Pohlmann, A. Speier: „Eine Diskussion über Trusted Computing – Sicherheitsgewinn durch vertrauenswürdige IT-Systeme“, IT-Sicherheit – Management und Praxis, DATAKONTEXT-Fachverlag, 5/2013
https://norbert-pohlmann.com/wp-content/uploads/2015/08/306-Eine-Diskussion-über-Trusted-Computing-Sicherheitsgewinn-durch-vertrauenswürdige-IT-Systeme-Prof-Norbert-Pohlmann.pdf

N. Pohlmann: "Cyber-Sicherheit – Das Lehrbuch für Konzepte, Mechanismen, Architekturen und Eigenschaften von Cyber-Sicherheitssystemen in der Digitalisierung“, ISBN 978-3-658-25397-4; 594 Seiten, Springer-Vieweg Verlag, Wiesbaden 2019
https://norbert-pohlmann.com/cyber-sicherheit/