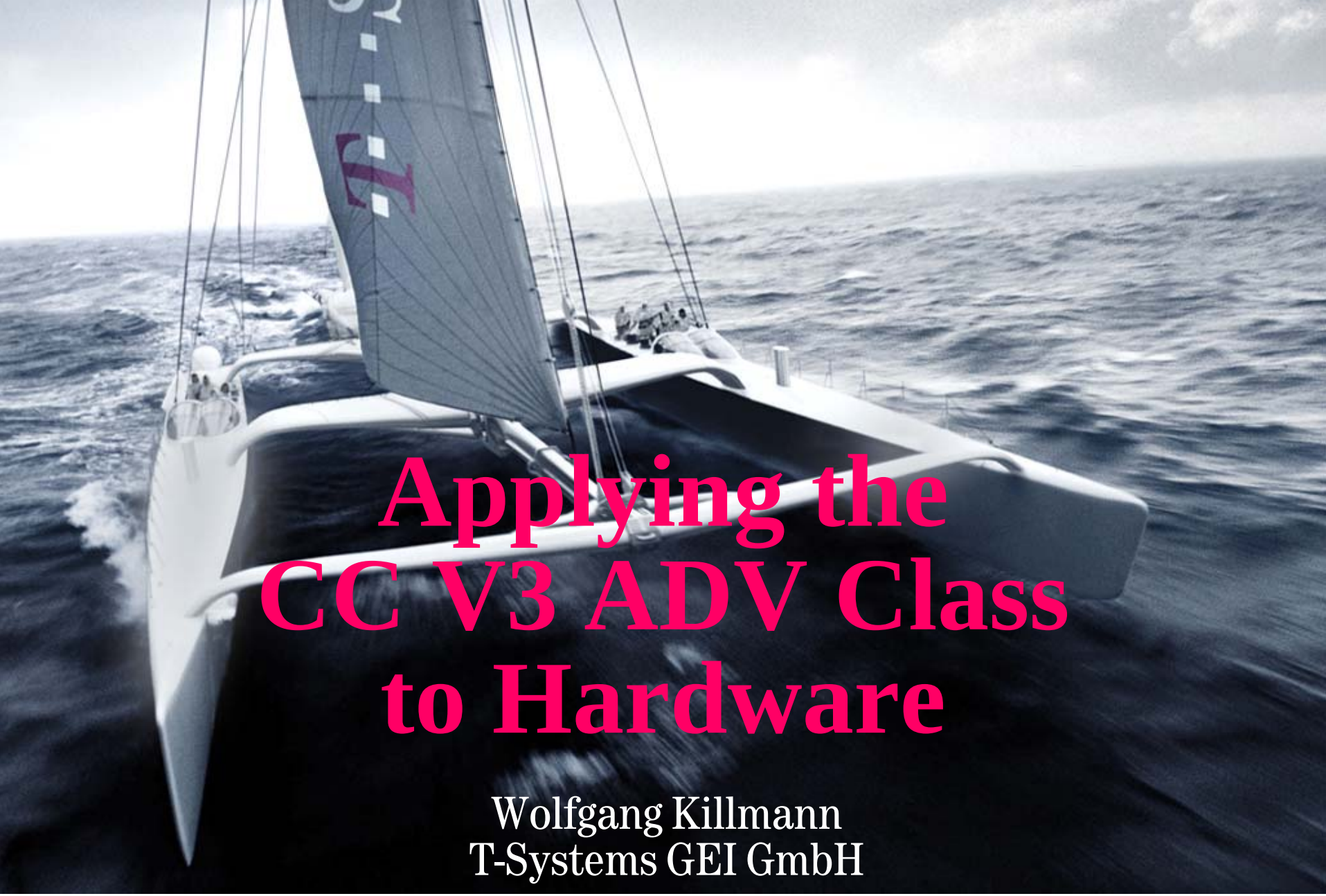


Applying the CC V3 ADV Class to Hardware

Wolfgang Killmann
T-Systems GEI GmbH

T Systems

The 7th International Common Criteria Conference
19-21 September, Lanzarote, Spain

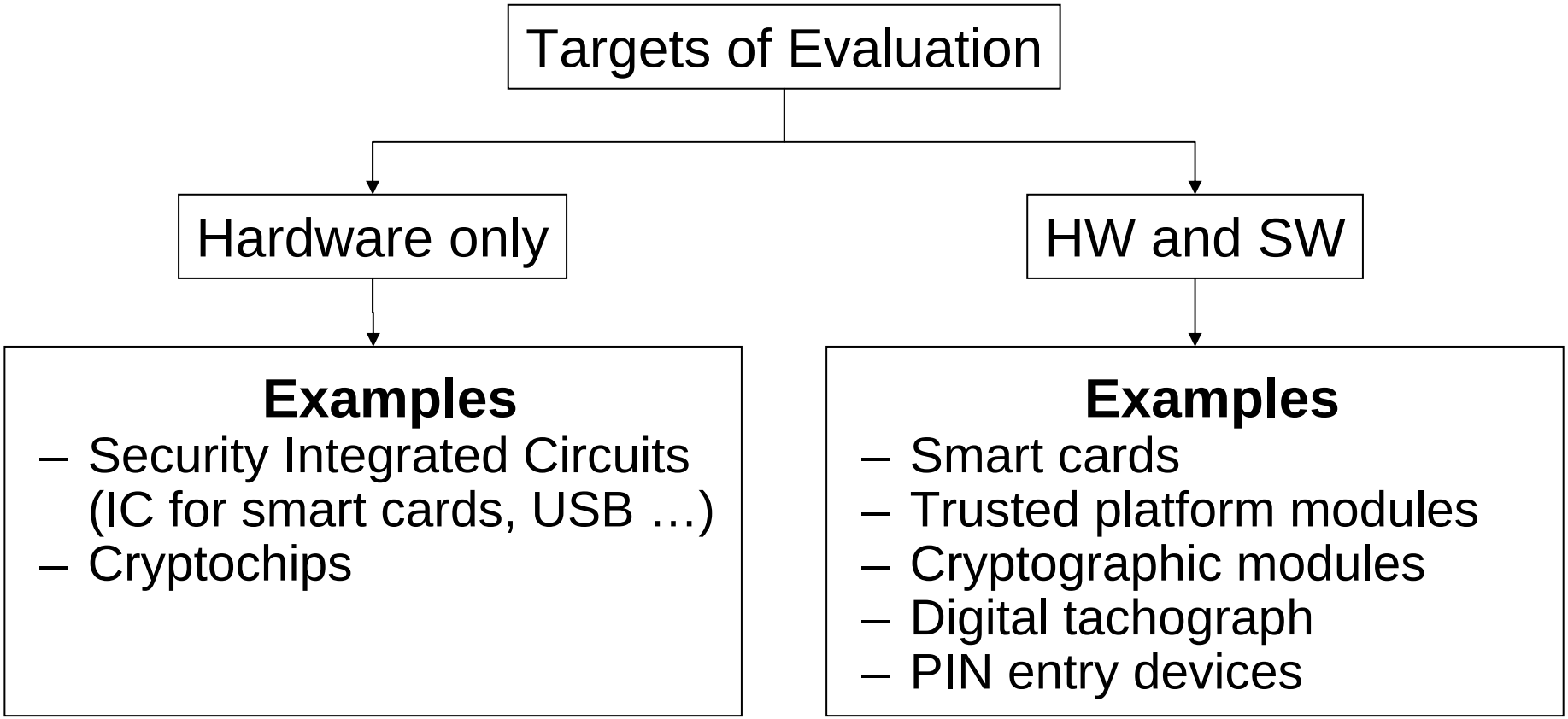
Motivation of the talk

The assurance class Development ADV was changed
- significant from CC version 2.3 to CC version 3.0
- to some extent from CC version 3.0 to CC version 3.1

CC and CEM explain the new aspects mainly for software TOE (e.g. ADV_TDS, ADV_ARC)

How do we apply the new ADV aspects to hardware?

Introduction
Target of Evaluation including hardware



A real IT system needs HW,
a TOE consists of HW or runs on HW platform or needs an OS running on HW

Introduction
Security Functional Requirements to Hardware

Security services provided by hardware

- Cryptographic operation FCS_COP
- Physical random number generators (FCS_RNG)
- Time stamps FPT_STM

TSF protection

- TSF physical protection FPT_PHP
- TSF self test FPT_TST
- Fail secure FPT_FLS, Fault tolerance FRU_FLT
- Internal TOE (TSF data) transfer FDP_ITT, FPT_ITT

SFR for HW in combination with SW

- Mainly the same but more complex
  e.g. coprocessor for modular arithmetic → RSA, ECC

Functional Specification ADV_FSP
TSFI and Ports

TSFI - consist of all means for users
(1) to invoke a service from the TSF (by supplying data that is processed by the TSF) and the corresponding responses to those service invocations or
(2) to affect the behaviour or the security of the TSF.

Port - a physical entry or exit point of the TOE that provides access to the TOE for physical signals, represented by logical interfaces, including power supply.

TOE ports shall be described and examined at least for
- side channel analysis (inherent or forced)
- fault induction analysis (perturbation)

Functional Specification ADV_FSP
TOE boundary

TOE physical boundary - an explicitly defined continuous perimeter that establishes the physical bounds of the TOE and contains all the hardware, software, and/or firmware components of the TOE.

Physical boundary shall be described and examined for

- physical protection (FPT_PHP)
- completeness of the port and interface description, including areas of emanation and for irradiation
- consistency of TOE architecture, design, and implementation

TOE Design ADV_TDS
General remark

CC v3 removed CC V2.3 requirements for information about “underlying hardware, firmware, and/or software required by the TSF“ and their supporting protection mechanisms

CC v3.1 gives more freedom to define subsystems and modules

CC v3.1 emphasizes rightly

- **categorization** as SFR-enforcing, SRF-supporting and SFR-non-interfering
- description of **behaviour** and **interaction** of subsystems
- **purpose** and **interfaces of modules**

TOE Design ADV_TDS
Algorithmic description

CC V2.3: The low-level design shall describe how each TSP-enforcing function is provided (ADV_LLD.2.6C).

CC V3.0: For each SFR-enforcing module, the design shall provide an algorithmic description detailed enough to represent the TSF implementation (ADV_TDS.3.15C).

CC V3.1: The design shall describe each SFR-enforcing module in terms of its purpose (ADV_TDS.3.7C).

TOE design shall describe these modules in terms of
- **What** does the module do (behaviour)
- **How (why)** does the module work (purpose)

so that determinations can be made about the **soundness** of the implementation of the SFRs

TOE Design ADV_TDS
Depth of description

	TSF Component			TSF Module		
	SFR Enforce	SFR Support	SFR NI	SFR Enforce	SFR Support	SFR NI
EAL4 ADV_TDS.3 Basic modular design (informal presentation)	description, interactions	description, interactions	description, interactions	common data, interfaces[2], algorithmic[3]	interaction, purpose	interaction, purpose
EAL5 ADV_TDS.4 Semiformal modular design (semiformal presentation)	description, interactions	description, interactions	description, interactions	common data, interfaces, algorithmic	common data, interfaces, algorithmic	interaction, purpose
EAL6 ADV_TDS.5 Complete semiformal modular design (semiformal presentation)	description, interactions	description, interactions	description, interactions	common data, interfaces, algorithmic	common data, interfaces, algorithmic	common data, interfaces, algorithmic

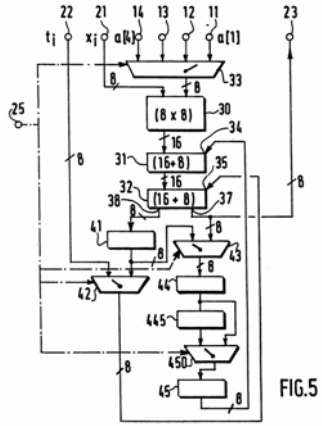
[3] *algorithmic* means an algorithmic description of the entire module is provided.

TOE Design ADV_TDS – Algorithmic Description
Example: Coprocessor for modular arithmetic

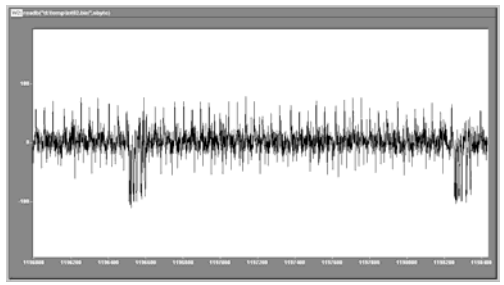
- Description of the modular arithmetic coprocessor
  → what modular arithmetic functions are implemented?
- How does the coprocessor work internally?
  → registers
  → logical operations
- Soundness of the coprocessor design
  → Does the execution time depend on values of the operands?

Example: Arithmetic function

$$x \leftarrow x \cdot y \bmod n$$



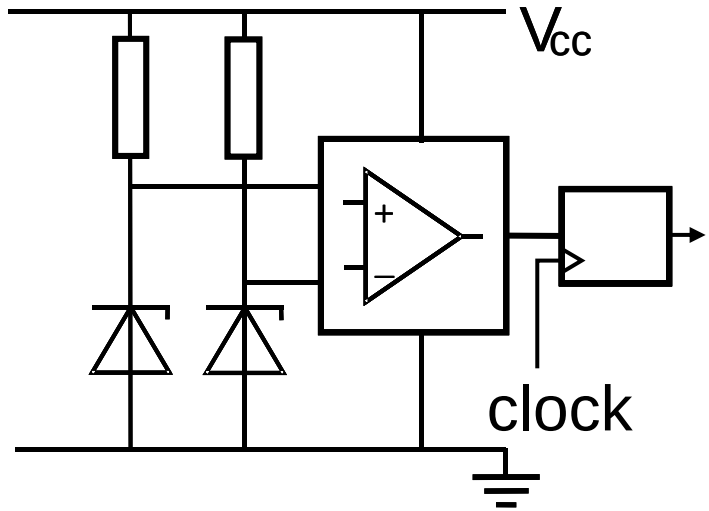
Picture: U.S. Patent 5,166,978



TOE Design ADV_TDS – Algorithmic Description
Example: Physical Random Number Generator

- Description of the physical random number generator
  → what are input and output?
- How does the RNG work
  → what is the source of randomness?
  → why is the source signal random?
  → how are the analogous signals digitized to bits?
- Soundness of the RNG design
  → What is the entropy of the output bits?
  → Stability of random signal (aging, environmental affected)?

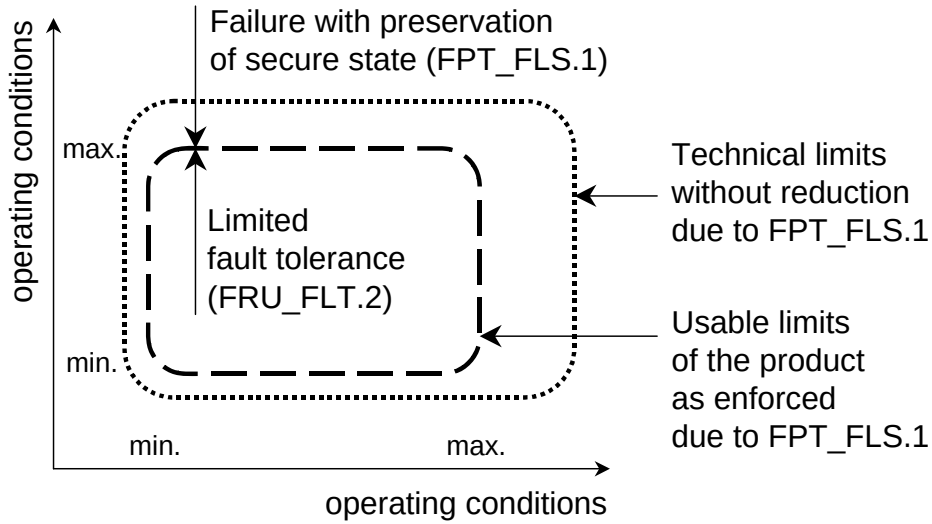
RNG with noise diodes (simplified scheme)



TOE Design ADV_TDS
Example: Environmental Control

- Description of environmental control
  → detection of violations
  → response to violations
- How do the sensors work
  → how does TOE detect violations of voltage, clock, temperature, … (physical principal of sensors)
  → robustness against light
- Soundness of the design
  → Does the TOE detect the thresholds outside which accurate function of the chip is not ensured

Example: Security IC PP (BSI-PP-0002)



TOE Design ADV_TDS versus ADV_ARC

All behaviour described in the ADV_TDS shall be mapped to the TSFIs that invoke it.
→ TSFI may be implemented by a port but may comprise also the whole physical boundary

Properties of addressed by ADV_ARC largely have no directly observable interface at the TSF.
→ criteria to distinguish between ADV_TDS and ADV_ARC

For hardware TSF self-protection, domain separation and non-bypassability is enforced by design and mechanisms.
→ analysis of TOE architecture and TOE design is closely related and dependent on each other.

Security Architecture ADV_ARC
General remark

TOE shall enforce the properties of self-protection, domain separation and non-bypassability properties for

- their own hardware TSF and
- their software running on the hardware platform.

Binding of hardware and software is critical to achieve these properties especially in case of composite evaluation, e.g.

- the software relays on hardware TSF like memory management for domain separation,
- the hardware may provide non-bypassability against physical attacks but may support side channel protection only.

Security Architecture ADV_ARC
Secure initialisation process

TSF initialisation is directed at the TOE components that are involved in bringing the TSF into an initial secure state (i.e. when all parts of the TSF are operational) when power-on or a reset is applied.

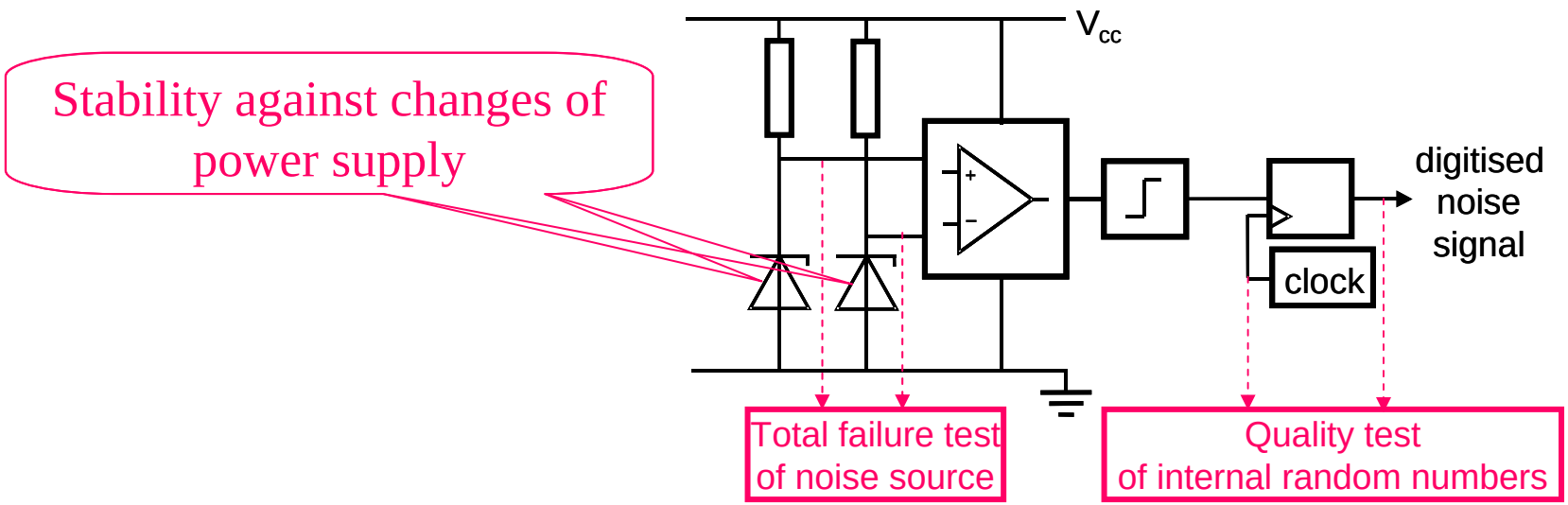
Example:

- power-on self test
  → operational mode detection
  → check of TOE integrity
  → self-test of Trusted platform module (TPM)
- sleep modes and weak-up procedures
  → sleep mode may disable some TSF (e.g. RNG) but keep other TSF active (e.g. sensors)

Security Architecture ADV_ARC
Self-protection from tampering

"**Self-protection**" - the ability of the TSF to protect itself from manipulation from external entities that may result in changes to the TSF.

Example: RNG with noise diodes



Security Architecture ADV_ARC
Security domains

Security domains - environments supplied by the TSF for use by potentially-harmful entities

Example:

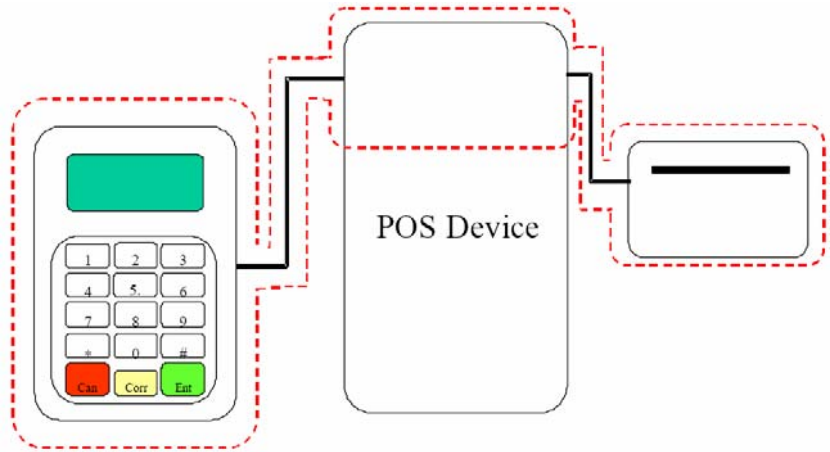
- privileged commands and resources
  → CPU protected mode
  → locality of TPM (registers, commands, priority)
- domain separation of memory
  → MMU separates memory used by trusted OS and untrusted applications
- power supply for key storage and environmental control independent on main power supply

Security Architecture ADV_ARC
Non-bypassability

Non-bypassability - property that the security functionality of the TSF (as specified by the SFRs) is always invoked.

Example:

physical tamper response
→ all over the physical boundary



Source of the Picture: APACS - PIN ENTRY DEVICE – PED GUIDELINE No 11, Type B2 – Individual PIN Pad & IC Reader to POS / EPOS

HW may be managed but can not be changed by SW
→ "last line of defence"

Conclusion

Application of CC version 3 class ADV to hardware requires specific guidance and interpretation for FSP, TDS and ARC

The supporting documents for security integrated circuits shall be updated based on the experience of practical evaluations