Composite Evaluation:

General Approach and Practical Integration of Security Policies

Dr. Igor Furgel

Volker Schenk

T-Systems ICT Security

What are we speaking about?

- Motivation
- Terminology and scope

- General approach (Composite Assurance Package)
- Assurance family ASE_COMP:
  "Coherence of composite product security policy"

- Practical Integration of Platform's Stipulations and Assumptions into Composite-ST

- Benefits of this approach

Motivation

- Final IT products consist of different (hard- and software) components being produced by different manufacturers
- The component manufacturers wish to keep the most possible independency from each other

Divide et impera!

- They try to use well-defined interfaces of different kinds: *technical*, *procedural*, *security*.
- A CC security certificate is a well-defined *security interface*.
- But how can we use it?

Motivation

- The aim of this contribution is to give

  – developers and
  – evaluators

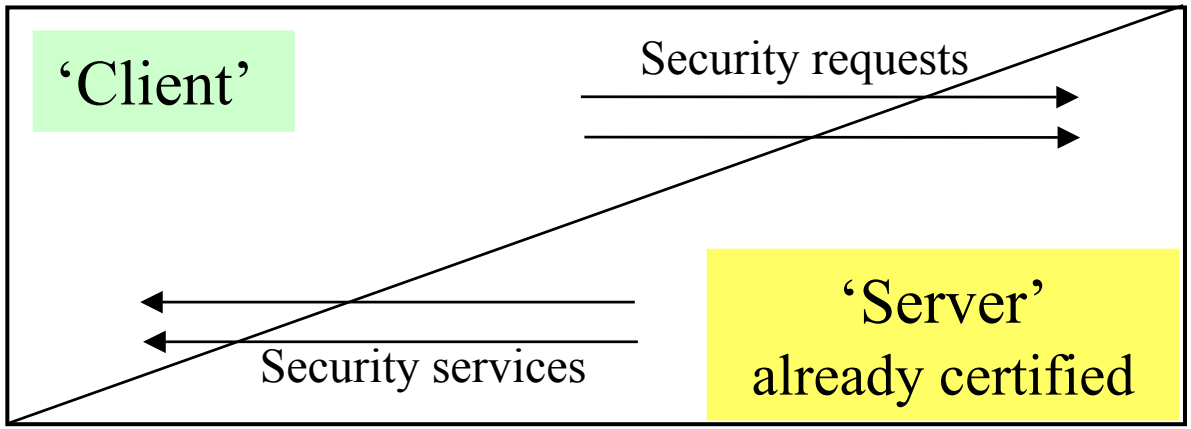
  a guidance

  -what relevant aspects have to be described and considered in the context of a composite evaluation and

  -how platform's stipulations / assumptions can be integrated into Composite-ST practically

- What is a *composite evaluation*?

Terminology & Scope



- A ***composite product*** consists of at least two different parts, whereby one of them represents a single product having already been evaluated/certified.
- The ***composite TOE*** comprises the whole composite product, i.e. the certified product is declared to be part of the composite TOE.
- An evaluation of the composite TOE is a ***composite evaluation.***

Terminology & Scope

- Usually, a composite product consists of two components, whereby the first one represents an ***underlying platform*** *('Server')* and the second one constitutes an ***application*** *('Client')* running on this platform. The underlying platform is usually the part of the composite product having already been evaluated.

	Smart card	**Java**	**Crypto-box**
application	Operating system	Java applet	Special crypto-box application (e.g. DigSign-Application)
underlying platform	Integrated circuit	Java run-time environment	Hardware + boot-loader + core operating system

General approach

- The most suitable type of the CC requirement constructs for the current aim is the assurance package: A package possesses an appropriate abstraction level being independent of concrete products and product families.
- We have defined (cf. ICCC5, 2004, Berlin)
  - a special assurance package for composite evaluation ***CompAP*** and
  - the evaluation methodology (evaluator actions) for this package.
- This methodology is independent of a CC version and thus applicable for CC v2.x as well as for CC v3.x.

General approach

- ***CompAP*** comprises the following assurance families:

ASE_COMP	Coherence of composite product security policy
ACM_COMP (v3.x: ALC_COMP)	Integration of composition parts
ADO_COMP (v 3.x: ALC_COMP)	Consistency of delivery procedures
ADV_COMP	Composite design compliance
ATE_COMP	Composite functional testing
AVA_COMP	Composite vulnerability assessment

The documents [ETR-LITE] and [ETR-LITE-ANNEX-A] were used as excitation for the assurance families of the *CompAP*, which is also compatible to them.

ASE_COMP: Coherence of Security Target - General methodology

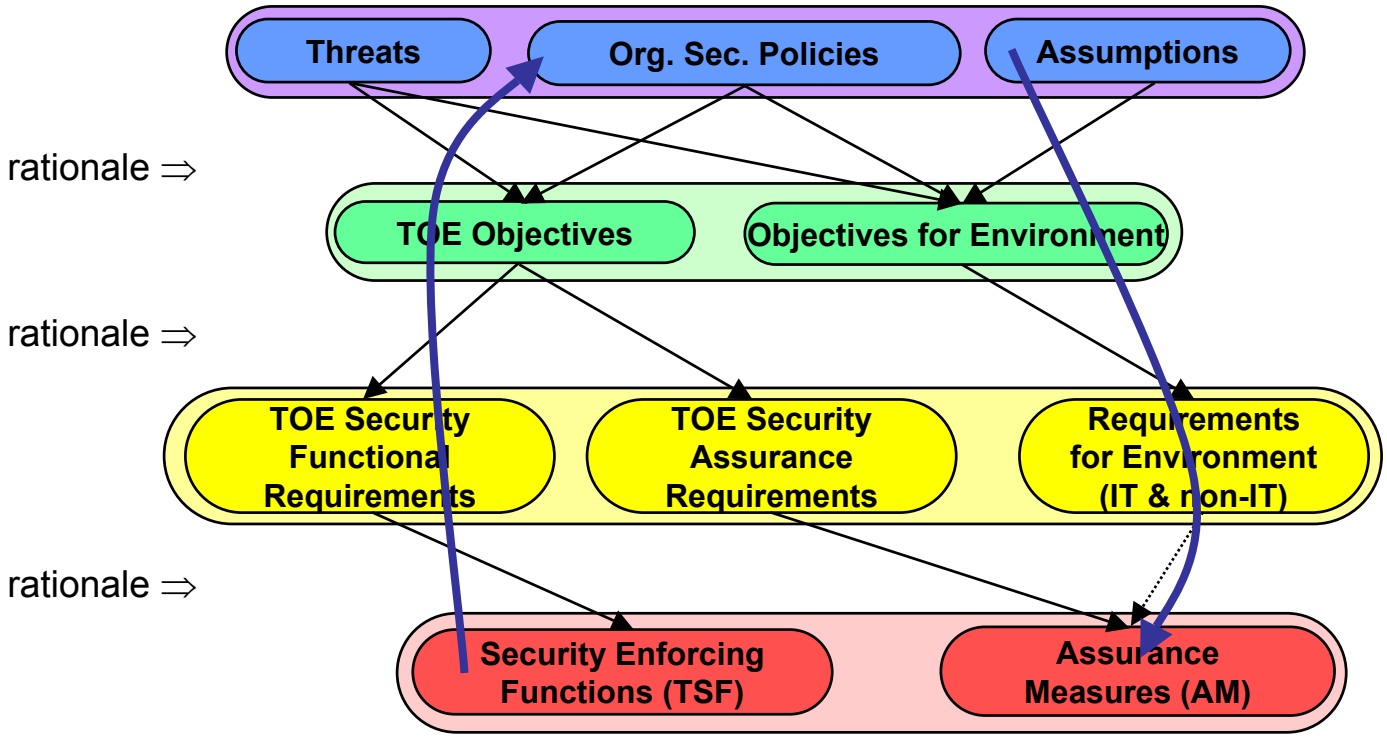
The aim of this component is to ensure that the security policy of the composite product does not contradict the security policy of the underlying platform.

'Three steps technology' for the ST:

- Step 1: The developer formulates a security policy of his composite product in form of a preliminary Security Target for the composite product using the standard code of practice. The Composite-SP can be formulated independent of the security policy of the underlying platform.
- Step 2: The developer determines the intersection of the Composite-SP and the Platform-SP by analysing and comparing their TSF.
- Step 3: The developer determines under which conditions he can trust in and rely on the Platform-TSF being used by the Composite-SP without a new examination.

ASE_COMP: Summary of the methodology

- Walk up-right-down through the structure of the Security Target of the platform



ASE_COMP: Summary of the methodology

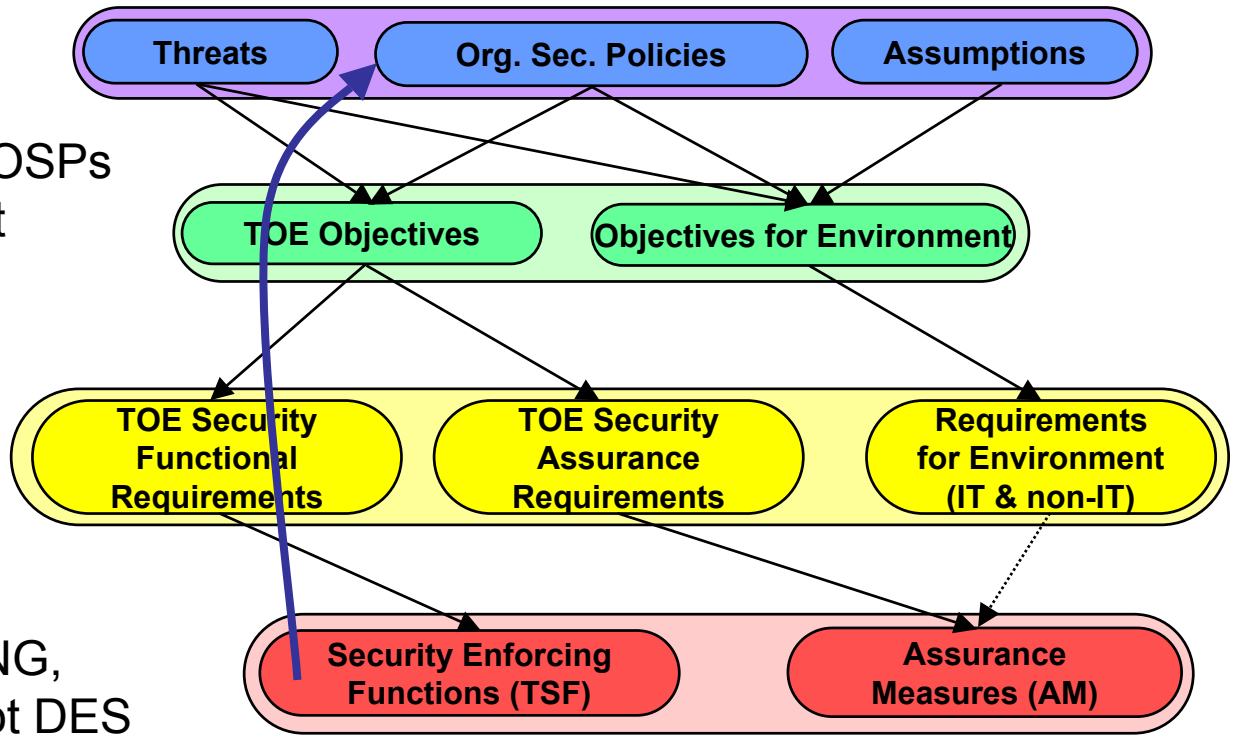
- Before you go up: Determine the intersection ***relevant PSF*** *(Platform Security Functions)* that have to be considered further:

- If the Composite-SP does not use any property of the Platform-SP and, hence, the intersection *relevant PSF* is an empty set, *no further composite evaluation activities are necessary*.
  In such a case there is a technical, but not a security composition.

ASE_COMP: Summary of the methodology

When you go up, consider only relevant items, i.e.

- only those TSF that use relevant platform security functions (PSF),
- only TSFR that are associated to relevant TSF,
- only TOE Objectives associated to relevant TSFR,
- and only threats and OSPs associated to relevant TOE Objectives.

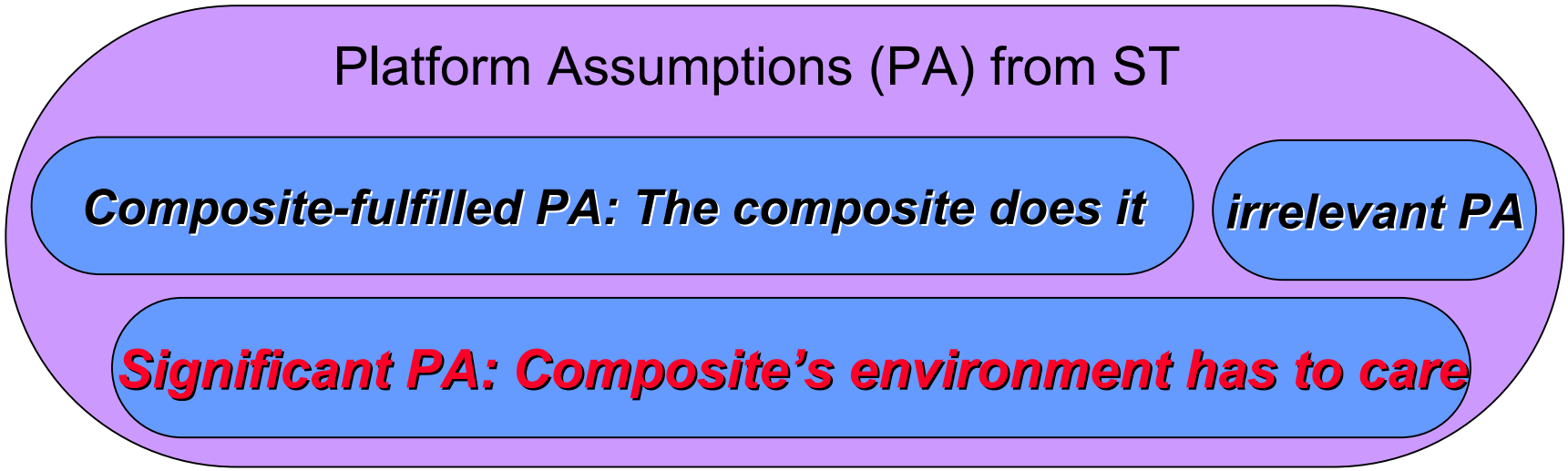


Example:

- smart card operating system on an integrated circuit card
- used HW features: RNG, AES, and RSA, but not DES

ASE_COMP: Summary of the methodology

- Before you go down: Determine the ***significant PA*** *(Platform Assumptions)* having to be considered further:



ASE_COMP: Summary of the Methodology

- How can I decide that the degree of trustworthiness of the relevant PSF (Platform Security Functions) is sufficient for the composite evaluation?

- I shall compare the Platform-AM (Assurance Measures) and the Composite-AM.

- The **degree of trustworthiness** of the Platform-TSF **is sufficient**, if

$$\text{Platform-AM} \supseteq \text{Composite-AM}$$

It is fulfilled, for example, if

$$\text{Platform-EAL} \supseteq \text{Composite-EAL}$$

Attention SOF.1:

$$\text{high} \supset \text{medium} \supset \text{basic}$$

Practical Integration of Platform's Stipulations and Assumptions into Composite-ST

The ST for the underlying platform usually defines

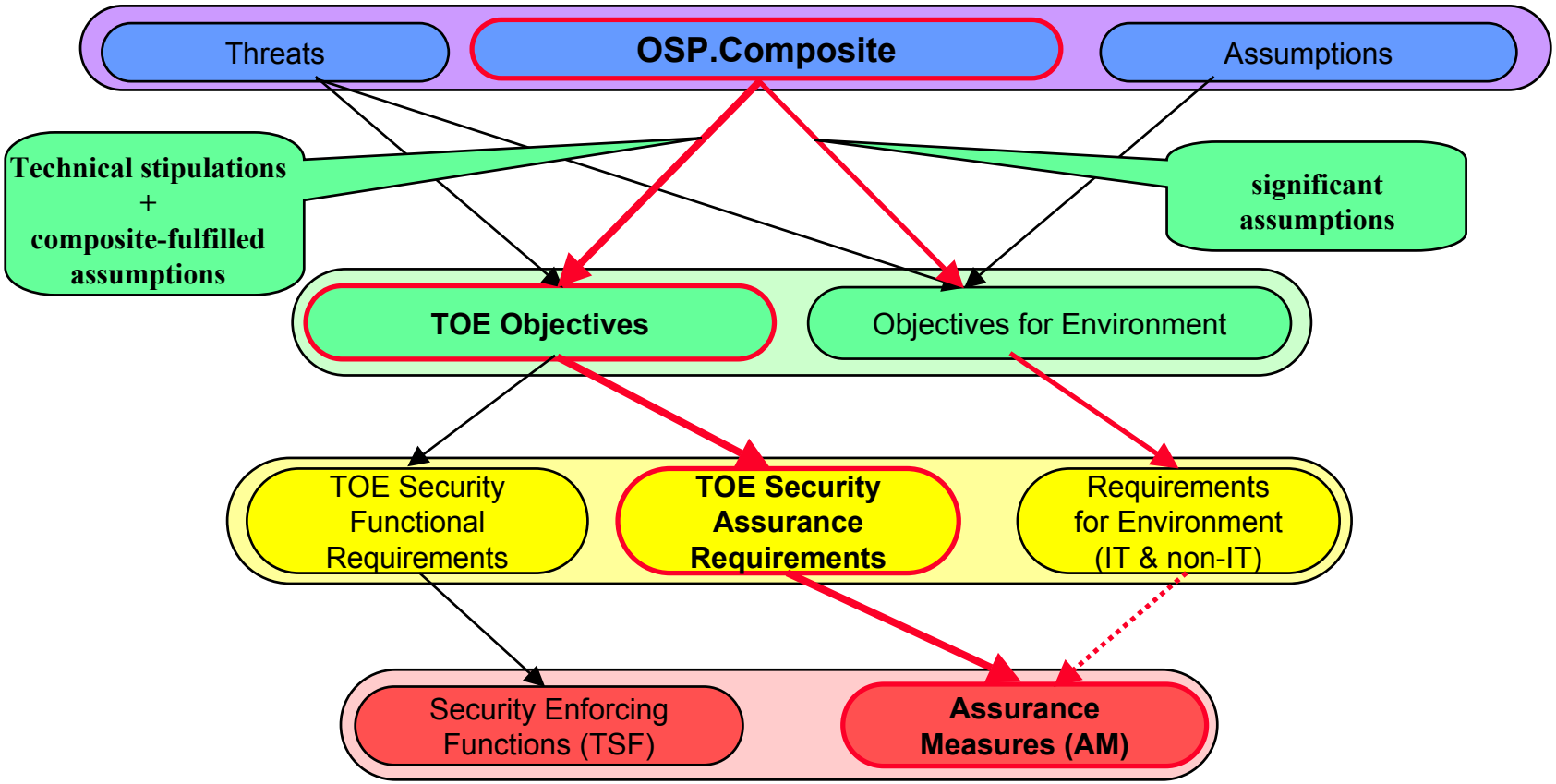
- several assumptions about the platform's environment.

The ETR-lite, certification report and user guidance usually contain

- additional stipulations – often of a technical nature – on the platform's environment.

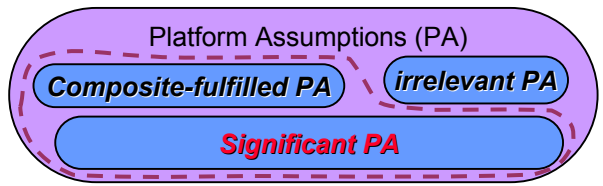
All composite-fulfilled and significant assumptions and relevant stipulations have to be reflected in the composite-ST.

Practical Integration of Platform's Stipulations and Assumptions into Composite-ST: Road Map



Practical Integration of Platform's Stipulations and Assumptions into Composite-ST – in only 5 moves

- **Move 1**: Define a dedicated policy **OSP.Composite**.
  The policy may sound like:
  "*The application (e.g. smart card OS) is running on a certified underlying platform (e.g. integrated circuit card) and is compatible to it, i.e. is respecting the platform's assumptions and stipulations.*"



- **Move 2**: List all composite-fulfilled and significant platform's assumptions about its environment (from the platform's ST) and stipulations on the platform's environment (from the platform's user guidance, ETR-lite and the certification report).

Practical Integration of Platform's Stipulations and Assumptions into Composite-ST – in only 5 moves

– **Move 3**: Define security objectives for every such assumption and stipulation.

a) For stipulations and composite-fulfilled assumptions, TOE objectives can always be formulated.

b) For significant assumptions, objectives for TOE's environment can always be formulated.

One or more assumptions and/or stipulations may be covered by one objective, if reasonable.

Practical Integration of Platform's Stipulations and Assumptions into Composite-ST – in only 5 moves

- **Move 4**: For every TOE objective, decide whether a functional or rather an assurance requirement fits better.
  From our experience, very often a refinement of an assurance requirement can cover a TOE objective,
  e.g. for ADO/ACM/ALC (v3.x: ALC), but also possible for ADV,
  e.g. ADV_LLD (v3.x: ADV_TDS) and ADV_IMP.

- **Move 5**: For every objective for the environment, formulate a requirement for the environment (either IT or non-IT).

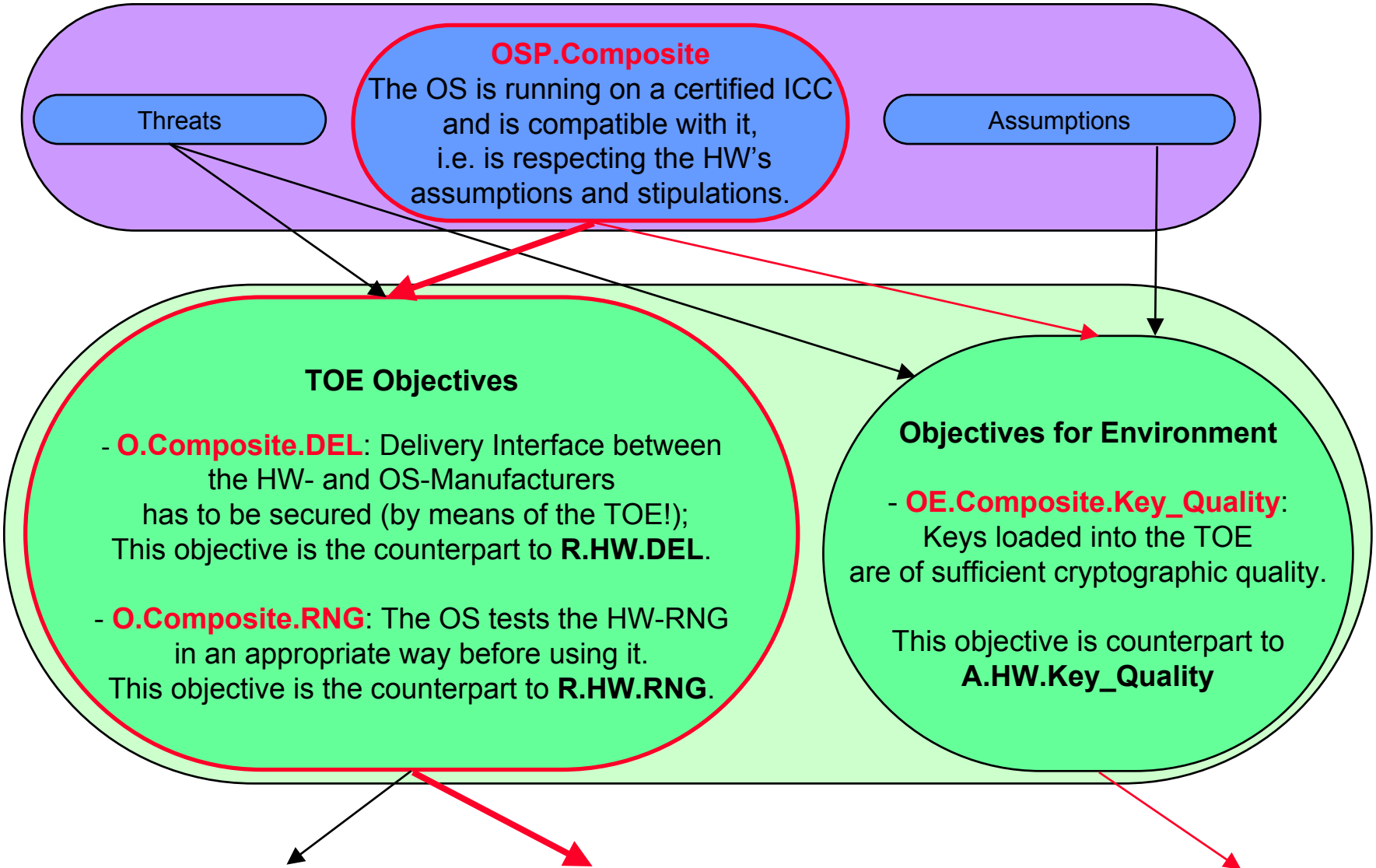
Practical Integration of Platform's Stipulations and Assumptions into Composite-ST: Example (1/4)

Example:
Smart card operating system building on a microcontroller

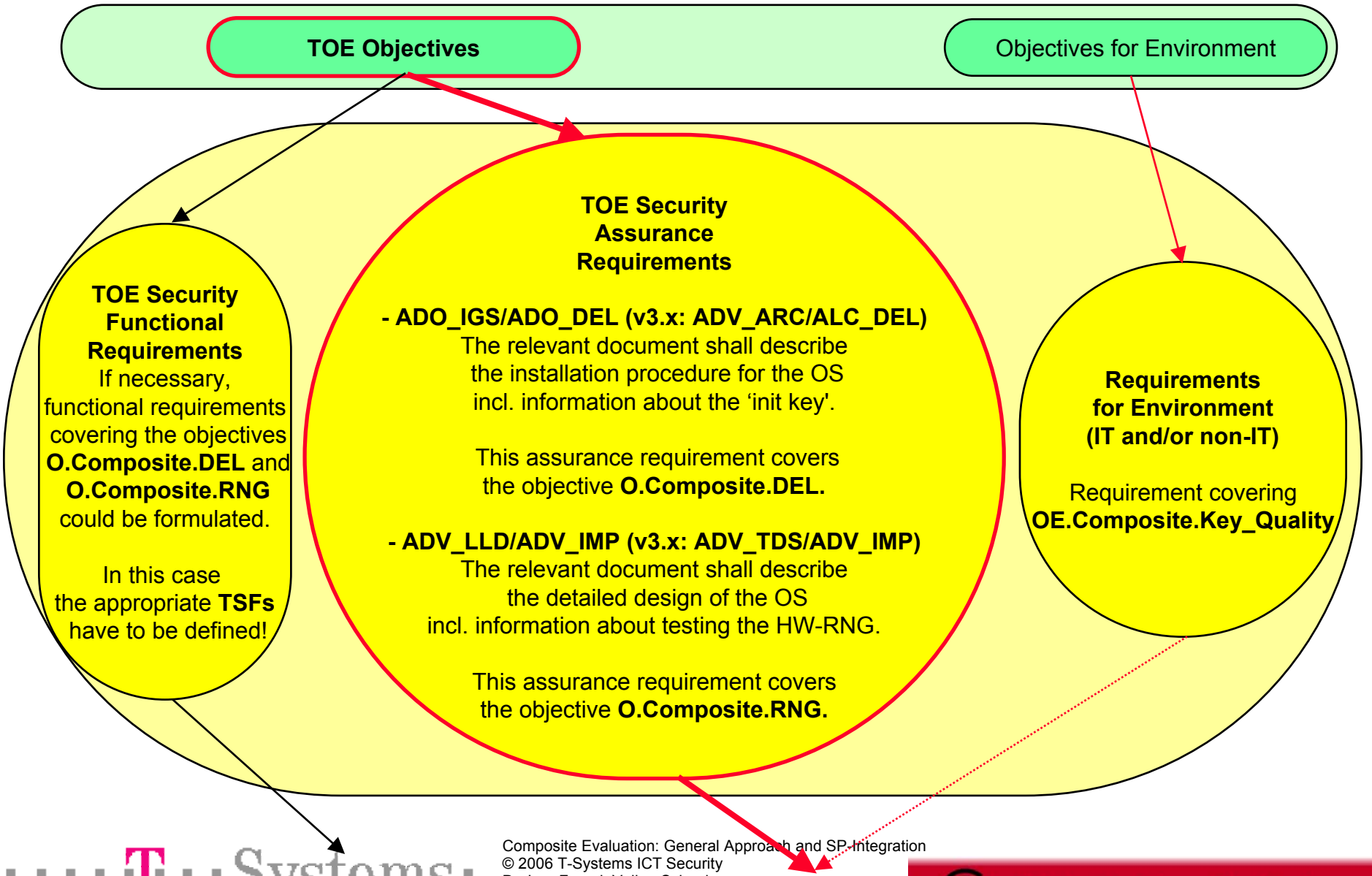
Let there be the following HW requirements and assumptions stated in the HW Certification Report, ETR-Lite and Guidance:

- **A.HW.Key_Quality**:
  Keys used are of sufficient cryptographic quality
- **R.HW.DEL**: OS has to be able to use an 'init-key' for securing delivery interfaces
- **R.HW.RNG**:
  OS has to perform appropriate tests before using the HW-RNG

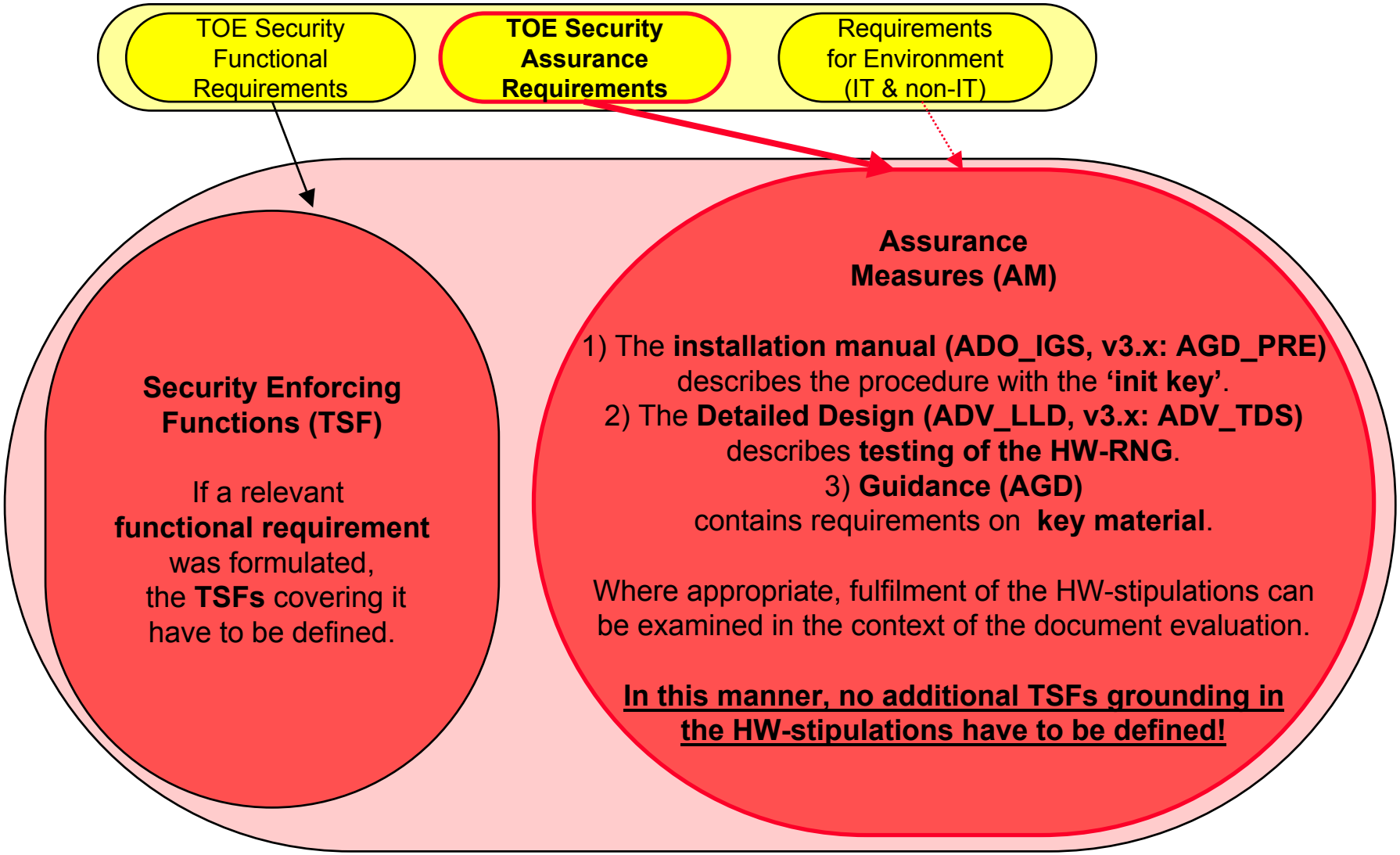
Practical Integration: Example (2/4)



Practical Integration: Example (3/4)



Practical Integration: Example (4/4)



Benefits of the Comp-AP approach (1/3)

- **Benefits**

  - Clear alignment with the **actual security features of the underlying platform** by justification of the composite product's Security Policy (relevant PSF, significant platform assumptions)
  - **Minimised risk** of getting incompatibility problems in a very late evaluation phase (e.g. vulnerability analysis or ETR), since compatibility is checked as early as possible
  - **Standardised approach** by definition of the composite assurance package and the methodology proposed
  - **Universally applicable**
    to all kinds of composite products and various CC versions

Benefits of the Comp-AP approach (2/3)

- **Benefits**
  - **Not every functionality** of the composite TOE necessarily has to be **raised to the status of a security function**.
    - If a refinement of an assurance component can do, the number of TSFRs and of TSFs will not grow uncontrolled.
  - **Improved transparency** of the security interoperability helps to eliminate the relevant composition flaws
  - **Improved quality**: clear concept and examination steps
  - **Fully compatible** with the approach in supporting document [ETR-LITE] and with the existing guidance [ETR-LITE-ANNEX-A]

Benefits of the Comp-AP approach (3/3)

- **Benefits**
  - **more confidence** in the security capability of a composite product for its user
  - **cost reduction** by excluding evaluated parts of a composite product.

Dr. Igor Furgel

T-Systems
ICT Security

Rabinstrasse 8
53111 Bonn

☎ +49 (228) 9841-512
igor.furgel@
t-systems.com

Volker Schenk

T-Systems
ICT Security

Rabinstrasse 8
53111 Bonn

☎ +49 (228) 9841-514
volker.schenk@
t-systems.com