Diffusion libre

Common Criteria

Evaluation des produits suivant les Critères Communs

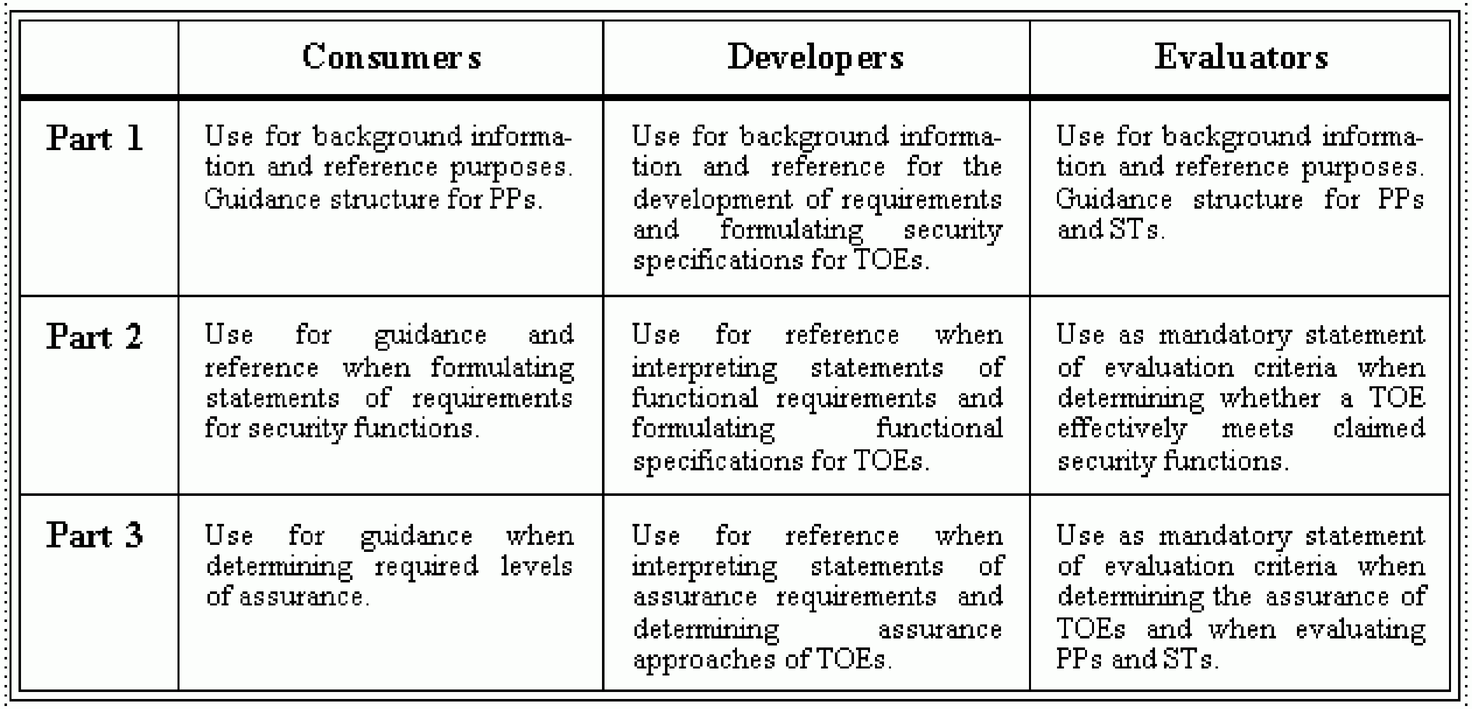
Alain MERLE (LETI/DSYS/CESTI)

References

- **CC : Common Criteria v2.1, August 1999**
  (http://www.scssi.gouv.fr/documents/docs/CC/cc21.html)
  - Part 1 : Introduction and general model
    August 1999 Version 2.1
  - Part 2 : Security functional requirements
    August 1999 Version 2.1
  - Part 3 : Security assurance requirements
    August 1999 Version 2.1

- **CEM : Common Methodology v1.0, August 1999**
  (http://www.scssi.gouv.fr/documents/docs/CEM/cem.html)
  - Evaluation methodology for PP and ST
    Version 1.0
  - Evaluation Methodology for levels EAL1 to EAL4
    Version 1.0

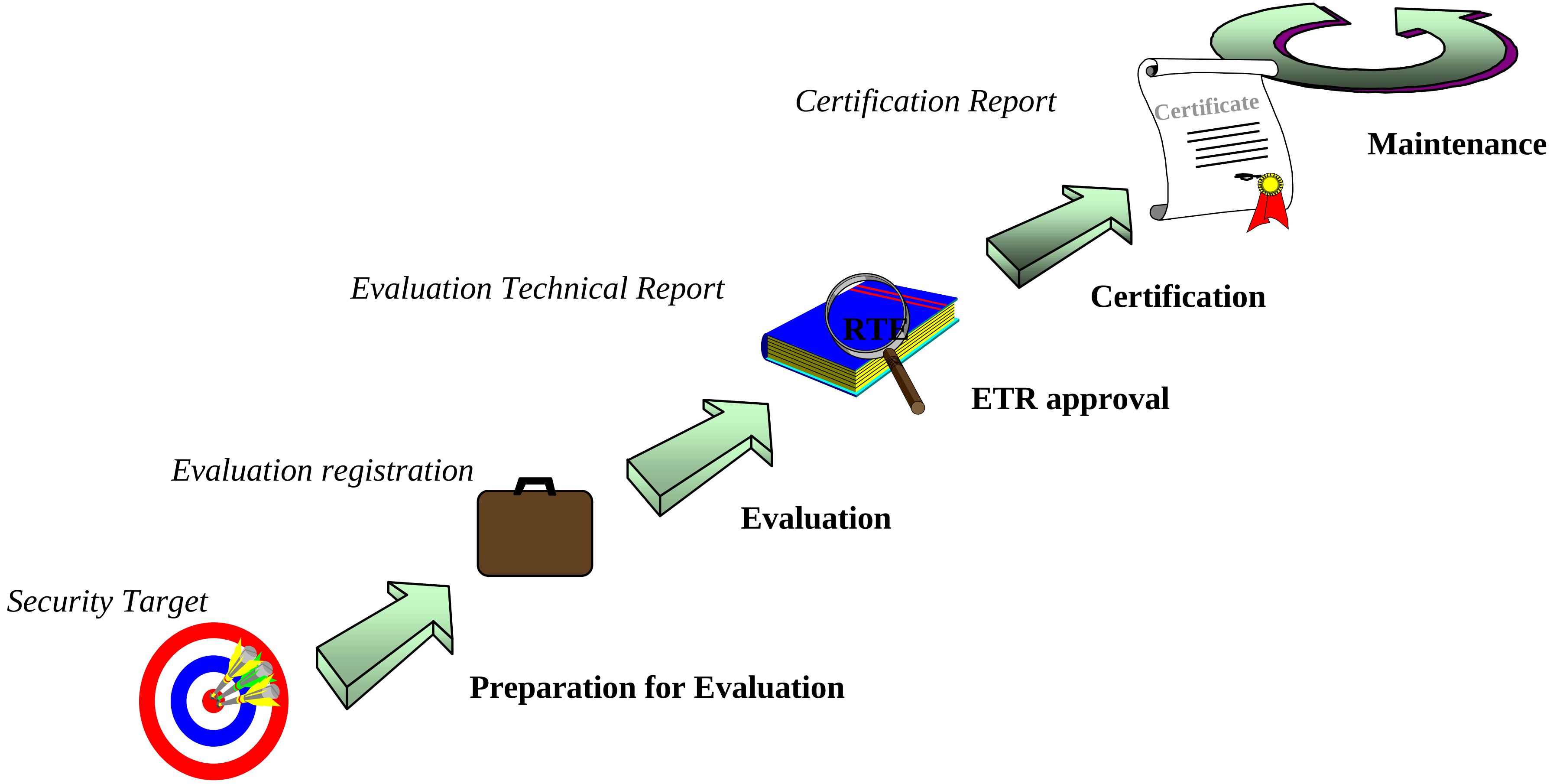
Roadmap to the Common Criteria



	Consumers	Developers	Evaluators
Part 1	Use for background information and reference purposes. Guidance structure for PPs.	Use for background information and reference for the development of requirements and formulating security specifications for TOEs.	Use for background information and reference purposes. Guidance structure for PPs and STs.
Part 2	Use for guidance and reference when formulating statements of requirements for security functions.	Use for reference when interpreting statements of functional requirements and formulating functional specifications for TOEs.	Use as mandatory statement of evaluation criteria when determining whether a TOE effectively meets claimed security functions.
Part 3	Use for guidance when determining required levels of assurance.	Use for reference when interpreting statements of assurance requirements and determining assurance approaches of TOEs.	Use as mandatory statement of evaluation criteria when determining the assurance of TOEs and when evaluating PPs and STs.

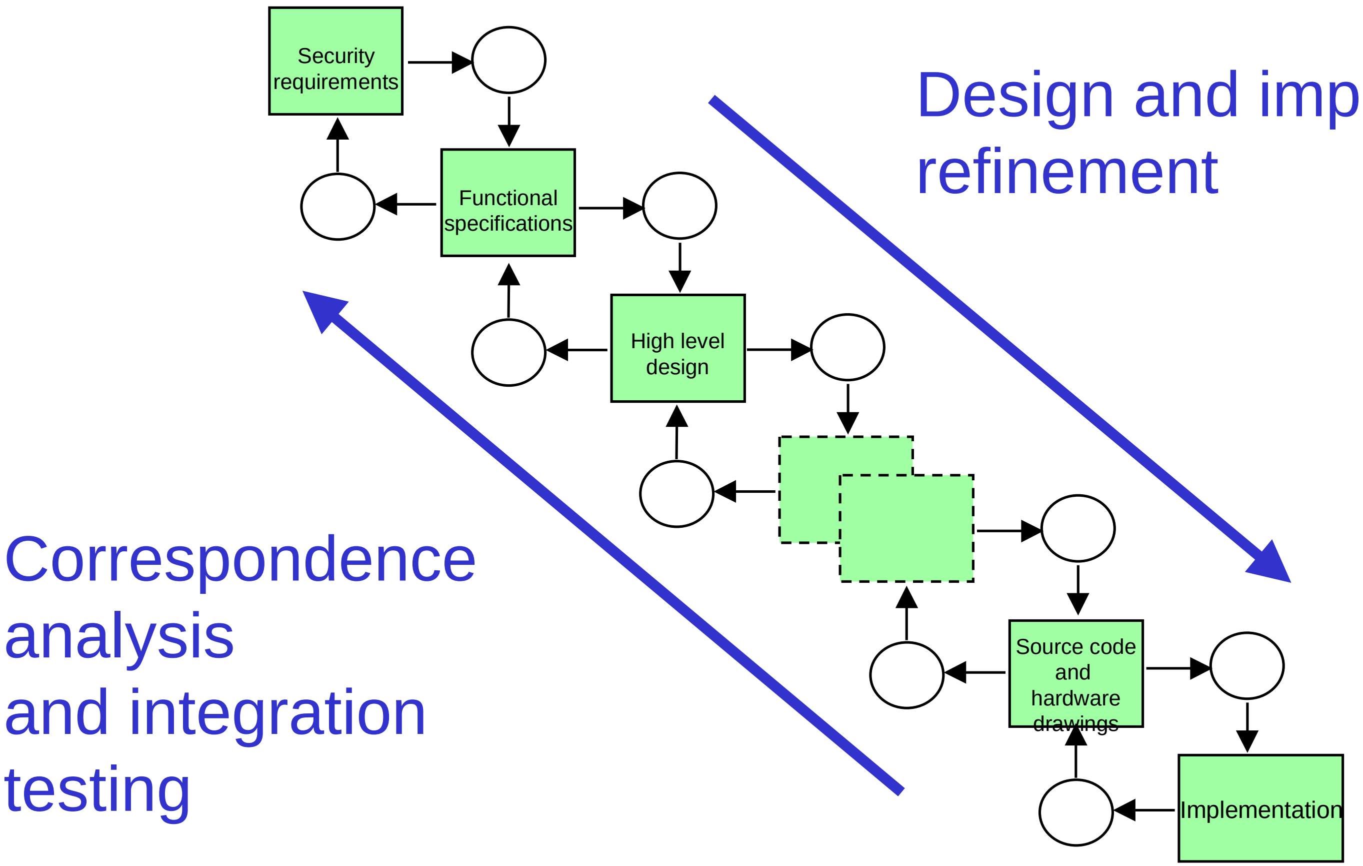
Evaluation Process : main steps

- .



Security Framework

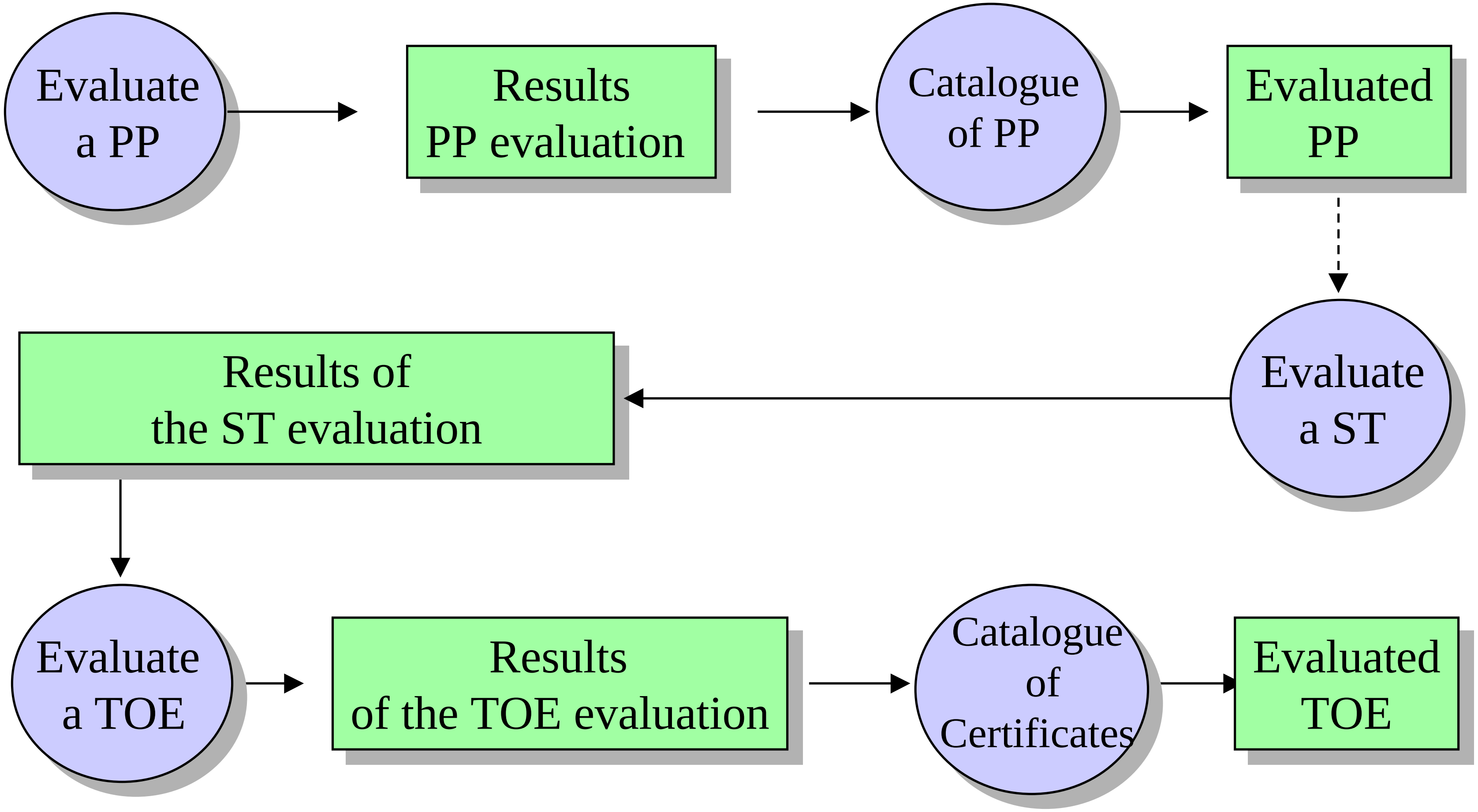
- .



Diffusion libre

CC evaluations and their results

-



Evaluate a PP

Results PP evaluation

Catalogue of PP

Evaluated PP

Evaluate a ST

Results of the ST evaluation

Evaluate a TOE

Results of the TOE evaluation

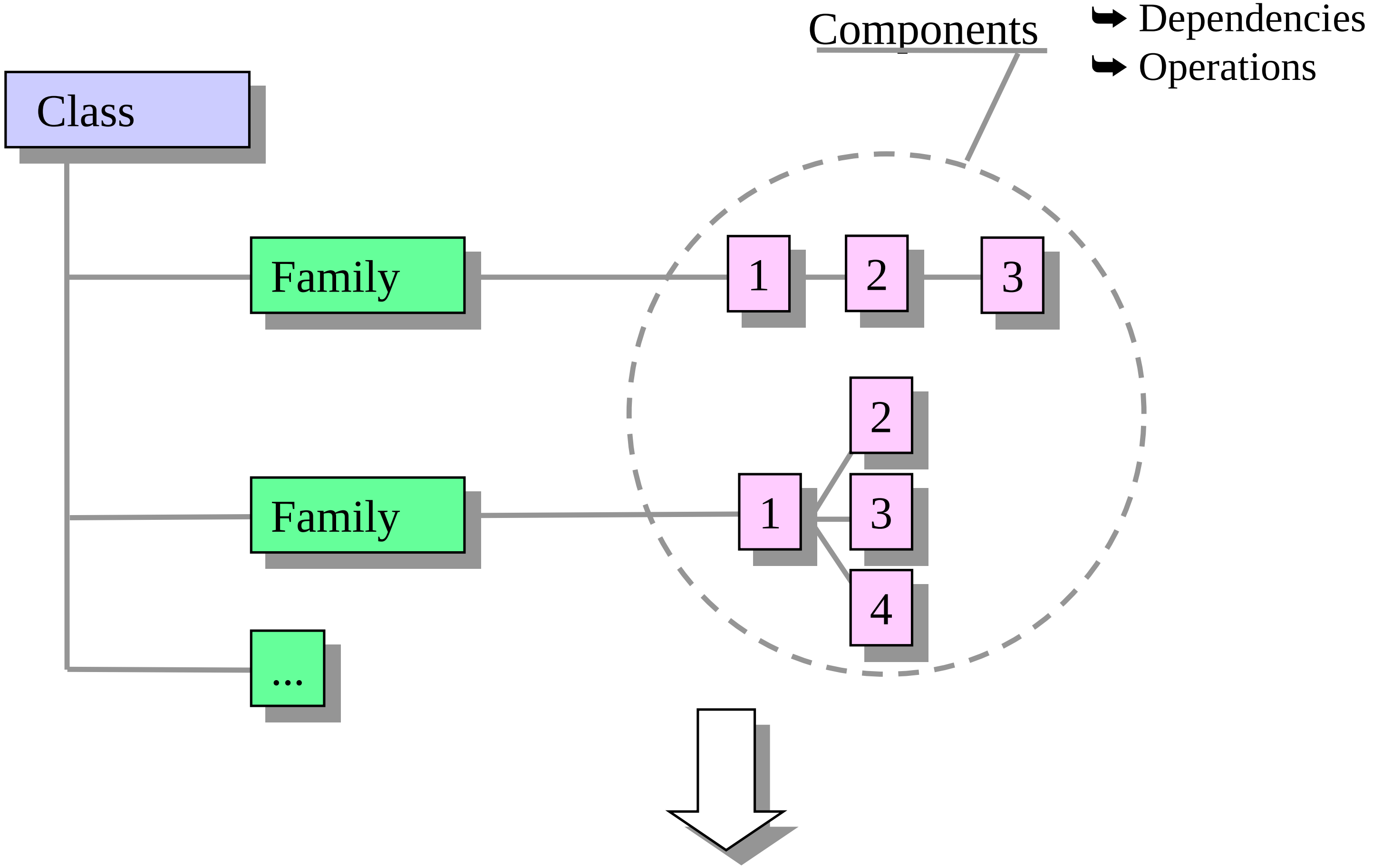
Catalogue of Certificates

Evaluated TOE

Key words : definitions

- Target of evaluation (TOE) : an IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation. It defines assets to protect.
- TOE Security Functions (TSF) : A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.
- TOE Security Policy (TSP) : A set of rules that regulate how assets are managed, protected, and distributed within a TOE.
- Security Target (ST) : Defines the target of evaluation, the environment, the threats, assets to protect, security objectives, assumptions.

Requirements : hierarchical structure

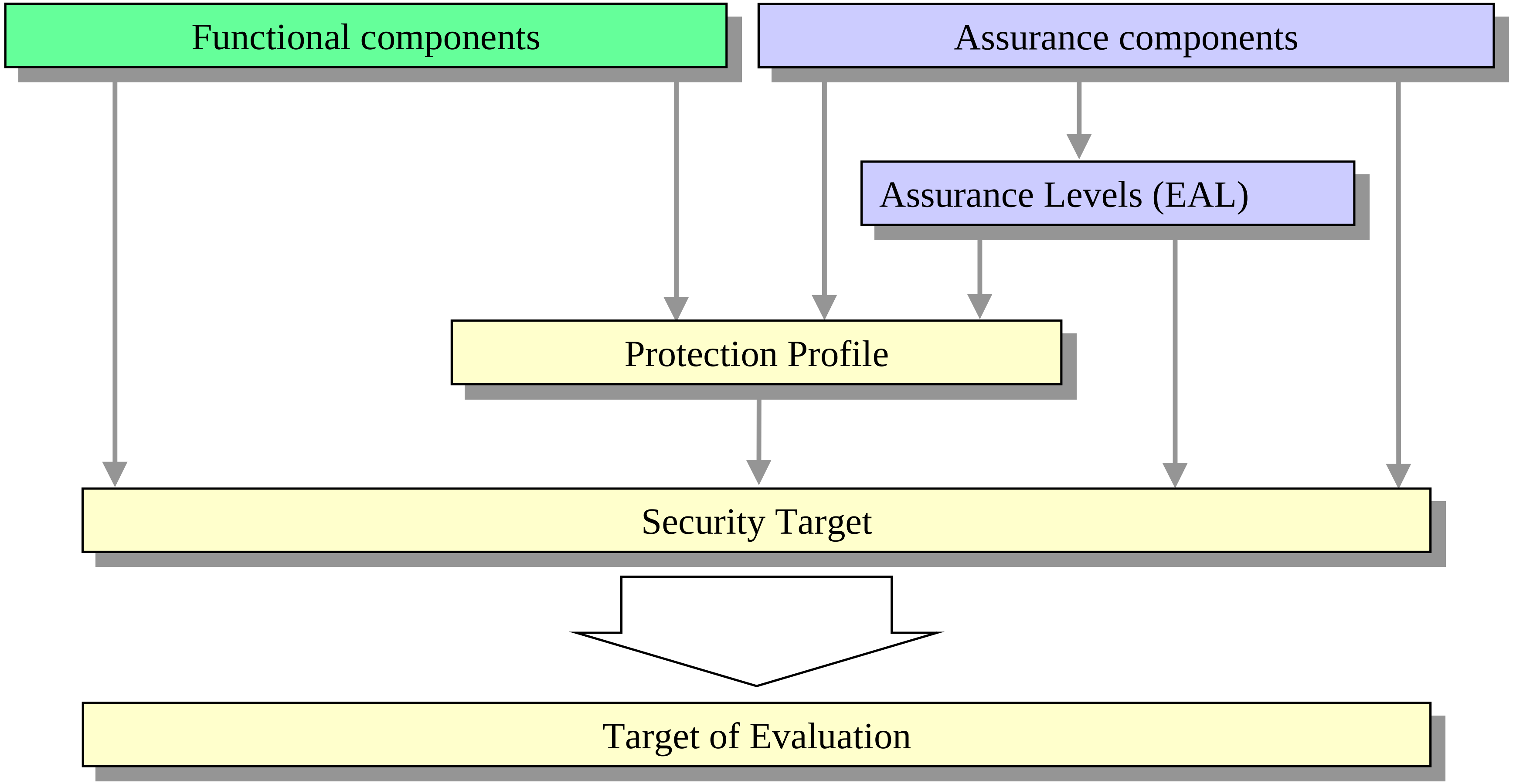


Applies to functional and assurance requirements

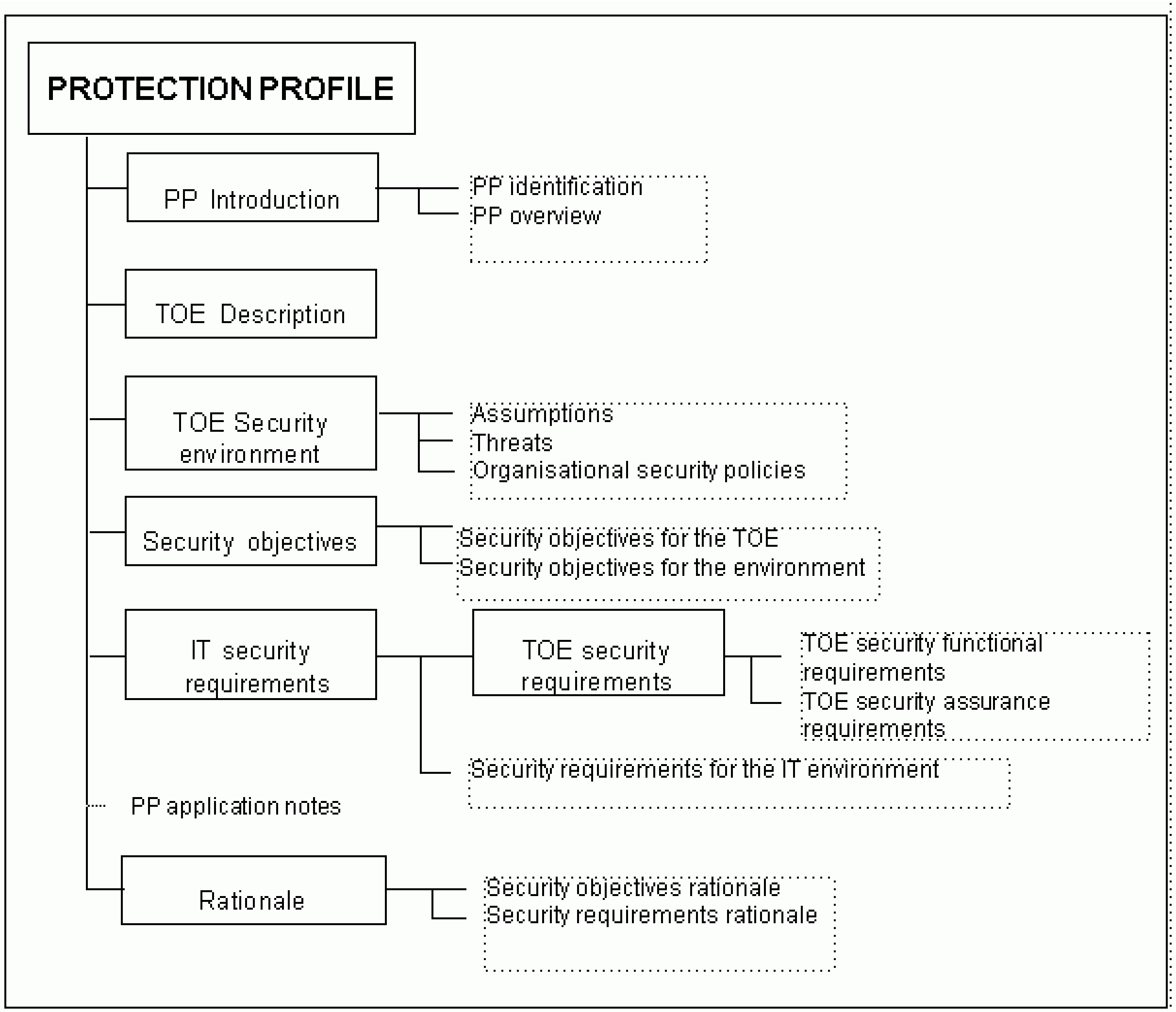
CC Catalogue : definitions

- **Class : A grouping of families that share a common focus.**
- **Family : A grouping of components that share security objectives but may differ in emphasis or rigour.**
- **Component : The smallest selectable set of elements that may be included in a PP, an ST, or a package.**
- **Package : A reusable set of either functional or assurance components (eg. an EAL), combined together to satisfy a set of identified security objectives.**

Use of Catalogue



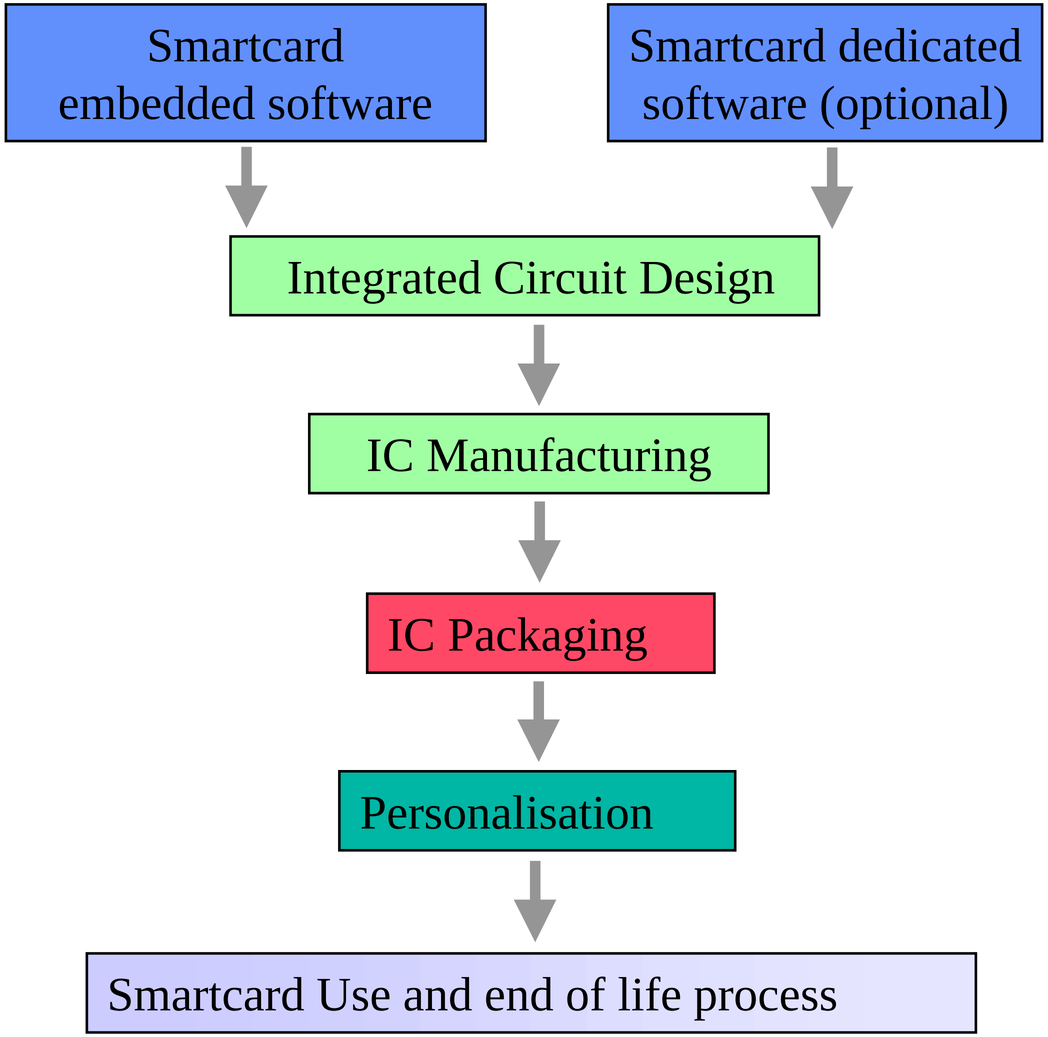
Protection Profile



PROTECTION PROFILE

- PP Introduction
  - PP identification
  - PP overview
- TOE Description
- TOE Security environment
  - Assumptions
  - Threats
  - Organisational security policies
- Security objectives
  - Security objectives for the TOE
  - Security objectives for the environment
- IT security requirements
  - TOE security requirements
    - TOE security functional requirements
    - TOE security assurance requirements
  - Security requirements for the IT environment
- PP application notes
- Rationale
  - Security objectives rationale
  - Security requirements rationale

PP Smartcards



Smartcard embedded software

Smartcard dedicated software (optional)

Integrated Circuit Design

IC Manufacturing

IC Packaging

Personalisation

Smartcard Use and end of life process

PP Embedded software PP/9809 ●

PP IEP ans PD PP/9909 ●

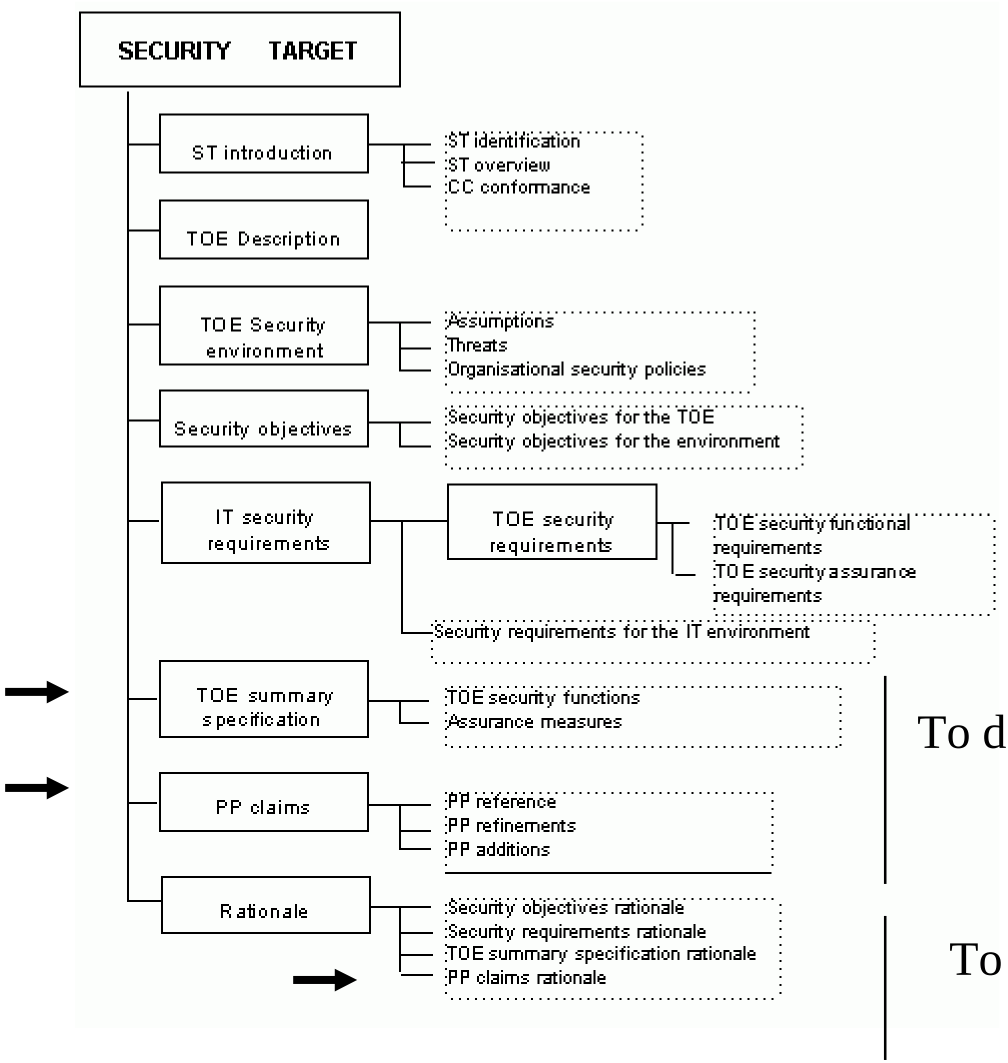
PP Integrated Circuit PP/9806 ●

PP IC packaging PP/9910

PP Personalisation

Security Target

- ▪



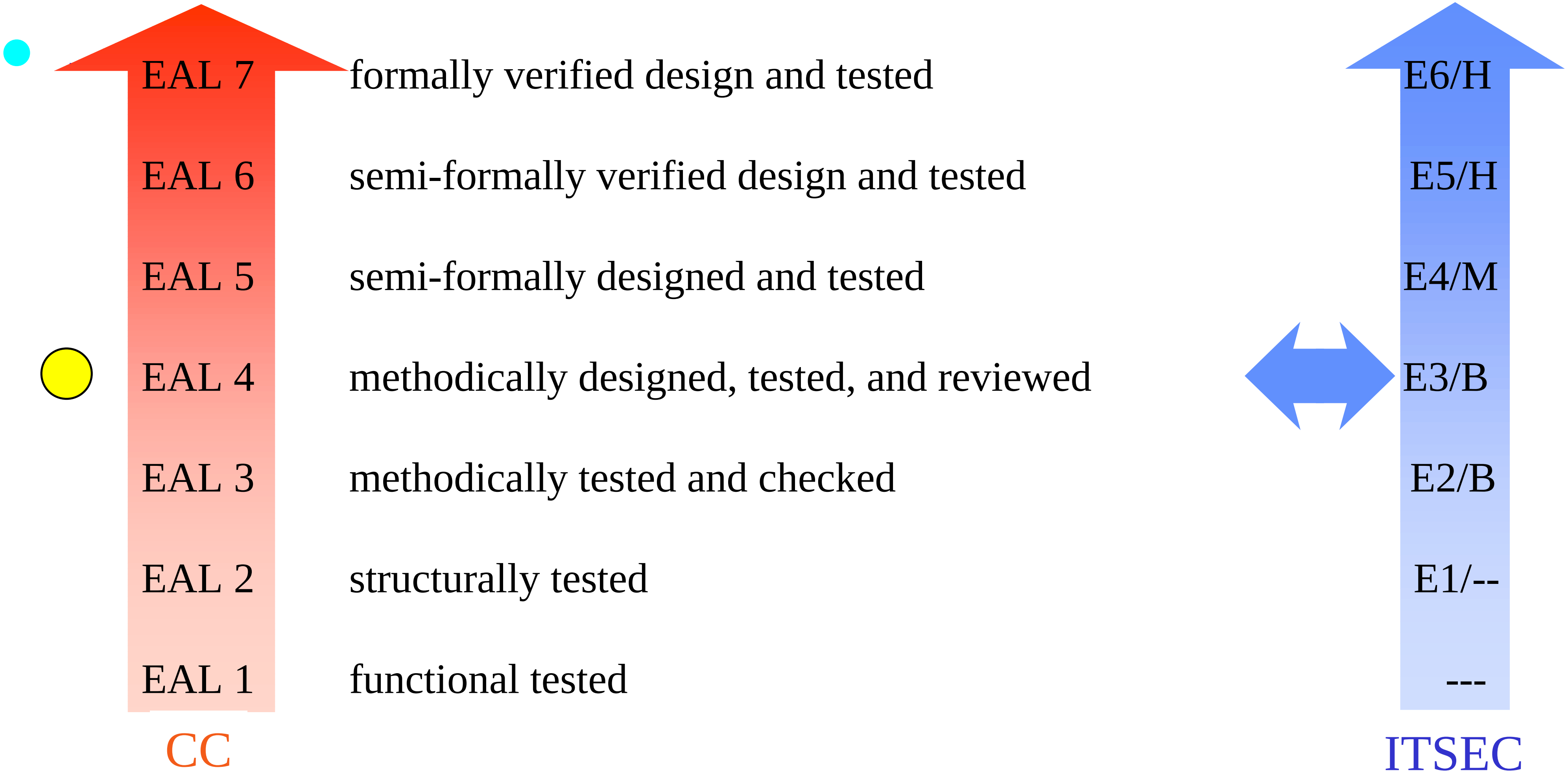
- SECURITY TARGET
  - ST introduction
    - ST identification
    - ST overview
    - CC conformance
  - TOE Description
  - TOE Security environment
    - Assumptions
    - Threats
    - Organisational security policies
  - Security objectives
    - Security objectives for the TOE
    - Security objectives for the environment
  - IT security requirements
    - TOE security requirements
      - TOE security functional requirements
      - TOE security assurance requirements
    - Security requirements for the IT environment
  - TOE summary specification
    - TOE security functions
    - Assurance measures
  - PP claims
    - PP reference
    - PP refinements
    - PP additions
  - Rationale
    - Security objectives rationale
    - Security requirements rationale
    - TOE summary specification rationale
    - PP claims rationale

To customize

To define

To define

Assurance level : 7 Levels



CC		ITSEC
EAL 7	formally verified design and tested	E6/H
EAL 6	semi-formally verified design and tested	E5/H
EAL 5	semi-formally designed and tested	E4/M
EAL 4	methodically designed, tested, and reviewed	E3/B
EAL 3	methodically tested and checked	E2/B
EAL 2	structurally tested	E1/--
EAL 1	functional tested	---

EAL 1

- *Could be used for an evaluation without the developer*

  – TOE security functions analysis
  – TOE functional specifications and interfaces
  – Security functions independent testing

EAL 2

- *Low-level independent evaluation*

  - TOE security functions analysis
  - TOE functional specification and interfaces
  - TOE sub-systems high-level design
  - Review of security functions black-box tests done by the developer
  - Obvious vulnerability assessment

EAL 3

- *Moderate level evaluation*

  - Grey-box testings
  - Independent confirmation of a selected sample of developer tests results
  - Search for vulnerabilities justified by the developer
  - Development environment control
  - TOE configuration management

EAL 4

- ***Complete white-box evaluation***
  - TOE modules low-level design
  - Subset of implementation representation
  - Independent search for vulnerabilities
  - Conformance of Development process against a life-cycle model
  - Tools identification
  - Automated configuration management

EAL 5

- ***High level of assurance obtained through a rigourous method of development***

  - All implementation analyses
  - Formal model and semi-formel presentation of functional specification and high-level design
  - Semi-formel Demonstration of correspondence
  - Modular design
  - Search for vulnerabilities and resistance to moderate potential attacks
  - Covert-channel analysis

EAL 6

High-level of risks product evaluation

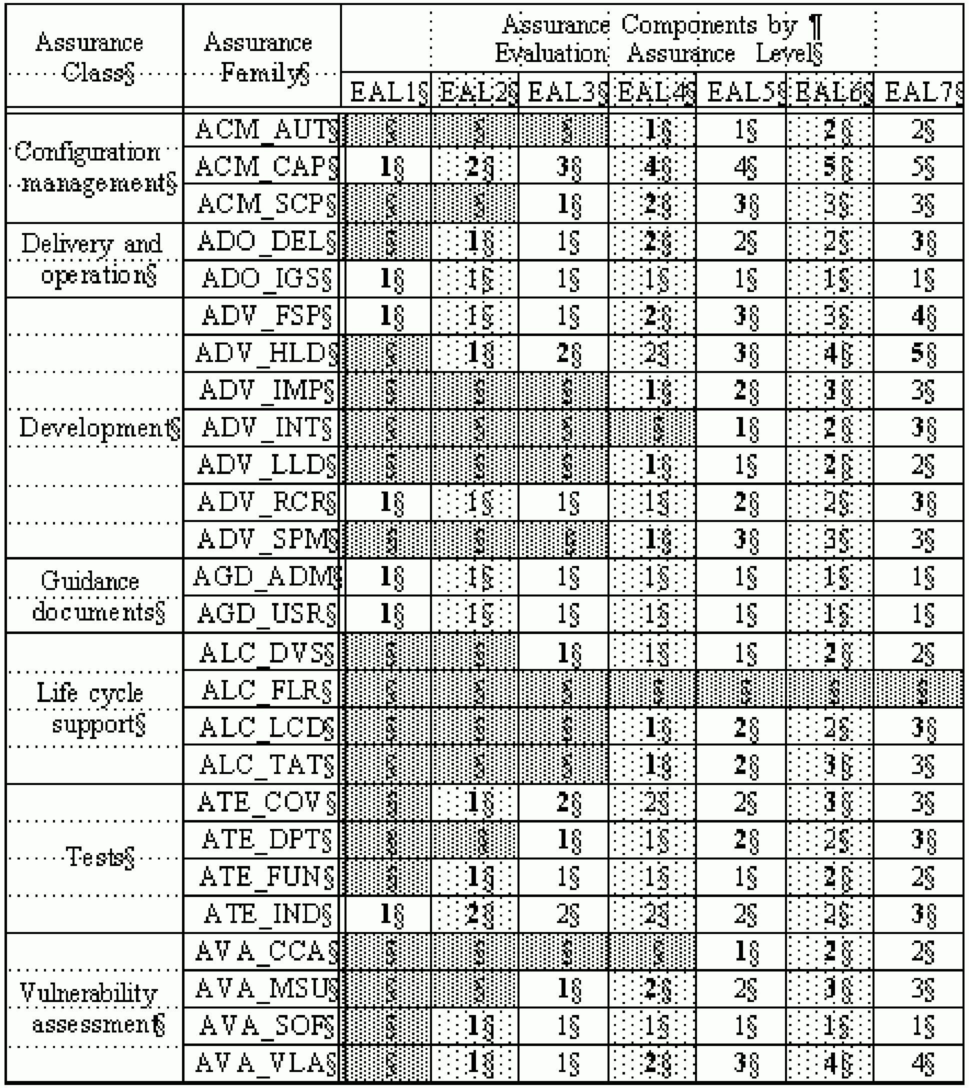
- Modular design and design by successive refinements
- Structured implementation of the TSF
- High-controlled development environment and advanced configuration management
- Systematic search for vulnerabilities and resistance to high potential attacks

EAL 7

- *Very-high level of risks product evaluation*
  - Formal presentation of functional specification and high level design
  - White-box developer Comprehensive testing
  - Complete and independent confirmation of developer tests results
  - Minimisation of design complexity

Evaluation Assurance level summary

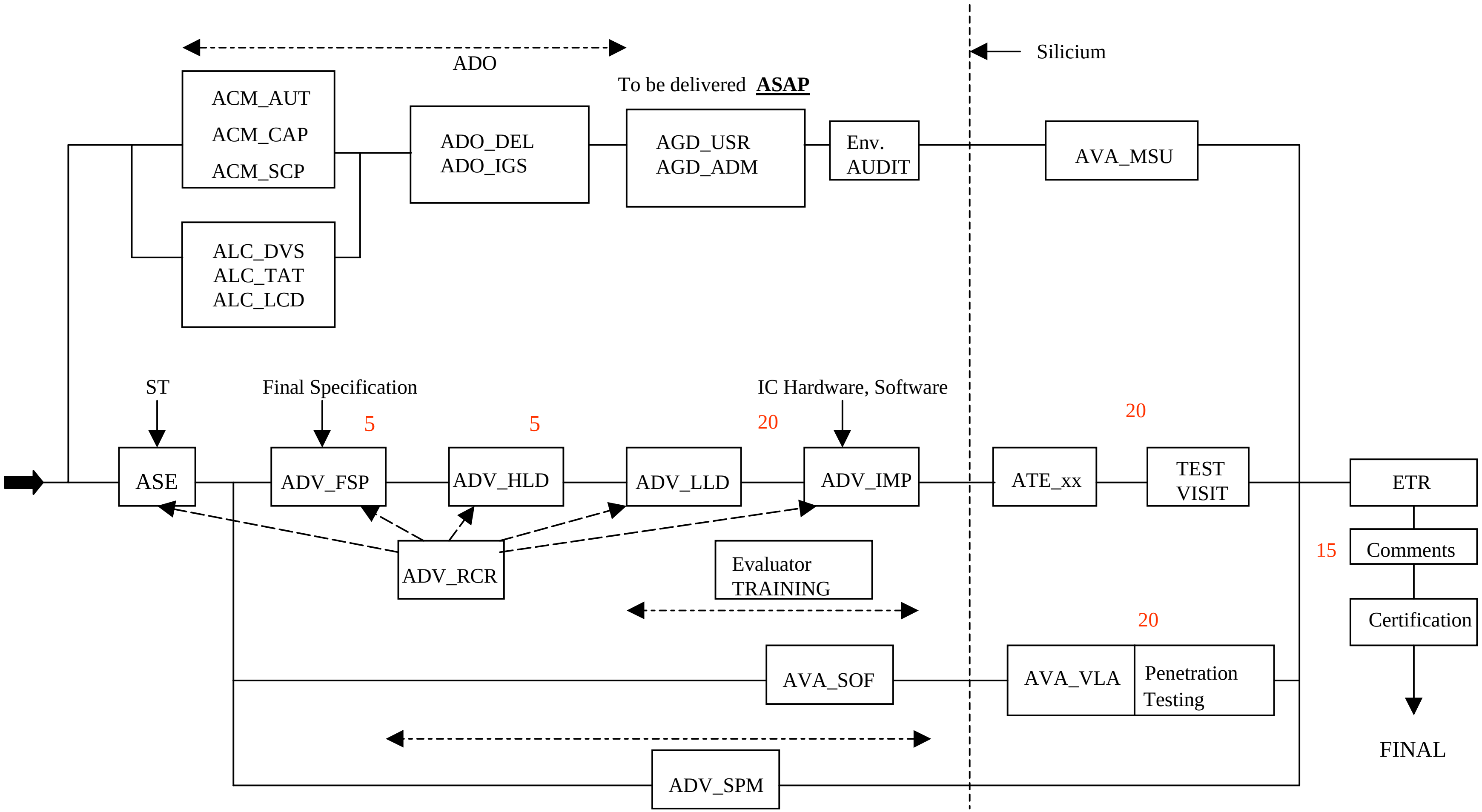
- .



Assurance Class§	Assurance Family§	Assurance Components by ¶ Evaluation Assurance Level§						
		EAL1§	EAL2§	EAL3§	EAL4§	EAL5§	EAL6§	EAL7§
Configuration management§	ACM_AUT§	§	§	§	1§	1§	2§	2§
	ACM_CAP§	1§	2§	3§	4§	4§	5§	5§
	ACM_SCP§	§	§	1§	2§	3§	3§	3§
Delivery and operation§	ADO_DEL§	§	1§	1§	2§	2§	2§	3§
	ADO_IGS§	1§	1§	1§	1§	1§	1§	1§
Development§	ADV_FSP§	1§	1§	1§	2§	3§	3§	4§
	ADV_HLD§	§	1§	2§	2§	3§	4§	5§
	ADV_IMP§	§	§	§	1§	2§	3§	3§
	ADV_INT§	§	§	§	§	1§	2§	3§
	ADV_LLD§	§	§	§	1§	1§	2§	2§
	ADV_RCR§	1§	1§	1§	1§	2§	2§	3§
	ADV_SPM§	§	§	§	1§	3§	3§	3§
Guidance documents§	AGD_ADM§	1§	1§	1§	1§	1§	1§	1§
	AGD_USR§	1§	1§	1§	1§	1§	1§	1§
Life cycle support§	ALC_DVS§	§	§	1§	1§	1§	2§	2§
	ALC_FLR§	§	§	§	§	§	§	§
	ALC_LCD§	§	§	§	1§	2§	2§	3§
	ALC_TAT§	§	§	§	1§	2§	3§	3§
Tests§	ATE_COV§	§	1§	2§	2§	2§	3§	3§
	ATE_DPT§	§	§	1§	1§	2§	2§	3§
	ATE_FUN§	§	1§	1§	1§	1§	2§	2§
	ATE_IND§	1§	2§	2§	2§	2§	2§	3§
Vulnerability assessment§	AVA_CCA§	§	§	§	§	1§	2§	2§
	AVA_MSU§	§	§	1§	2§	2§	3§	3§
	AVA_SOF§	§	1§	1§	1§	1§	1§	1§
	AVA_VLA§	§	1§	1§	2§	3§	4§	4§

Diffusion libre

Evaluation Flow



Common Criteria

EAL4 + Evaluation

- Erwan Froc (CNET/SVA/e-CB - LETI/DSYS/CESTI)

V1.0 : 01/12/99
V1.1 : 06/01/00

Plan

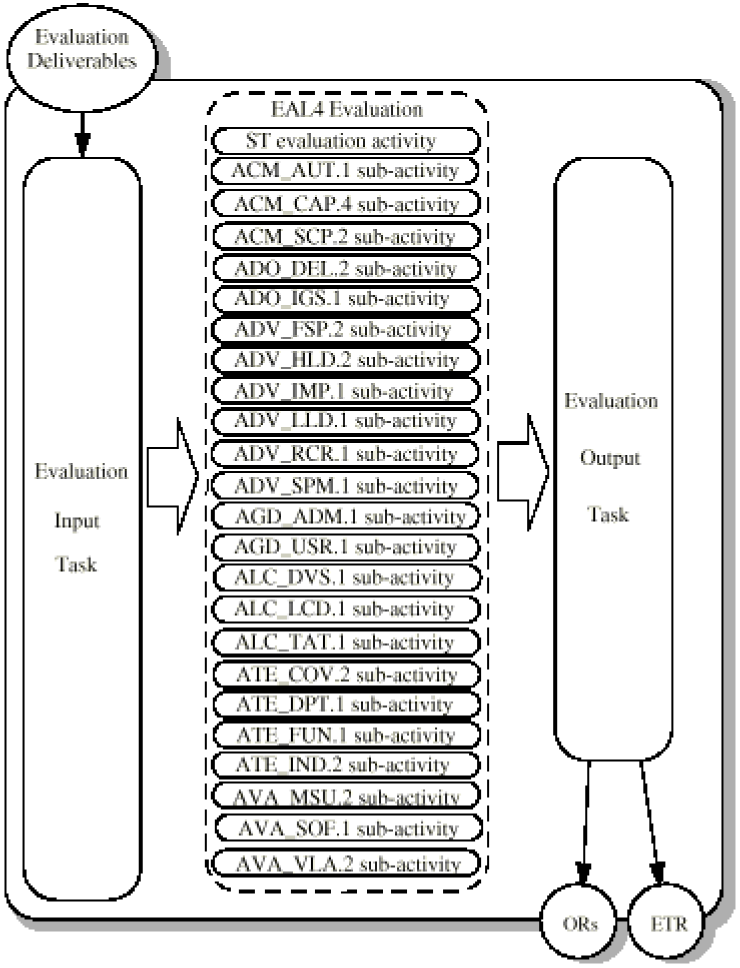
- **EAL 4 Evaluation activities** ←
  - Objectives
  - Input documents
  - Assurance requirements
- **Maintenance activities**

EAL4 evaluation relationships

- **Activities to conduct a complete EAL4 evaluation :**
  - Evaluation input task
  - EAL4 evaluation activities comprising the following:
    - evaluation of the ST
    - evaluation of the configuration management
    - evaluation of the delivery and operation documents
    - evaluation of the development documents
    - evaluation of the guidance documents
    - evaluation of the life cycle support
    - evaluation of the tests
    - testing
    - evaluation of the vulnerability assessment
  - Evaluation output task

Task, activities and subactivities for an EAL4 evaluation

- .



ACM : Configuration Management Activity

- **ACM_AUT.1 : Evaluation of CM automation**
- **ACM_CAP.4 : Evaluation of CM capabilities**
- **ACM_SCP.2 : Evaluation of CM scope**

⇒ **An audit is scheduled to verify described procedures**

ACM_AUT.1 - CM automation

- **Objective**
  - to determine whether changes to the implementation representation are controlled with the support of automated tools, thus making the CM system less susceptible to human error or negligence.
- **Input**
  - Configuration management documentation

ACM_AUT.1 : CM automation

- **Use of a Configuration Management Plan**
  - Tool documentation
  - Used procedures

- **Provide this CM Plan (applied to the product under evaluation)**
  - Automated CM system
    - Control access (prevent unauthorized modification)
    - Prove that only authorized changes are made in the :
      - implementation representation
  - Automated means to support the generation of the TOE (make files…)
  - Description of the automated tools used in the CM system
  - Description how these tools are used

ACM_CAP.4 - CM capabilities

- **Objective**
  - to determine whether a CM system is used to preserve the integrity of the TOE throughout its development and maintenance.
  - This helps ensure that the evaluation results are not undermined as a result of unauthorised changes being made to the evaluated version of the TOE or the configuration items of which it is comprised.

- **Input**
  - TOE suitable for testing
  - Configuration management documentation

ACM_CAP.4 - CM capabilities

- **Provide a reference for the TOE**
  - Uniquely referenced (product, items and documentation)
  - TOE labelled with its reference
  - Description of the method used to uniquely identify the configuration items
- **Developper shall use the CM system**
  - Description of how this CM system is used
  - Demonstration : CM system is operating in accordance with CM plan
- **Developper shall provide CM system documentation**
  - Configuration list
    - TOE, reticule, design, layout, configuration files...
  - CM system
    - all configuration items shall be maintained under CM system
    - measures : only authorised changes are possible to the configuration items
  - Acceptance plan

ACM_SCP.2 - CM scope

- **Objective**
  - to determine whether as a minimum the developer performs configuration management on the :
    - TOE implementation representation,
    - design, tests,
    - user and administrator guidance,
    - the CM documentation and security flaws.

- **Input**
  - List(s) of configurations

ACM_SCP.2 - CM scope

- **Provide CM documentation**
  - The configuration list shall include :
    - TOE implementation representation (hardware and software)
    - Design documentation
    - Test documentation
    - User and administrator documentation
    - CM documentation
    - Security flows
      - (e.g. problem status reports derived from a developer ’s problem reporting database)
  - Documentation shall show how the status of each configuration items can be tracked throughout the CM documentation

ADO Delivery and operation activity

- **ADO_DEL.2 : Evaluation of delivery**
- **ADO_IGS.1 : Installation and operation activity**
  - **⇒ An audit is scheduled to verify decribed procedures**

ADO_DEL.2 - Evaluation of delivery

- **Delivery procedures (physical or electronic):**
  - TOE & associated tools
  - Design, libraries, firmware
  - Files to the maskshop and reticules
  - Design, Libraries, Firmware
  - ROM code
  - Documents relative to the TOE
- **It shall contain :**
  - Identifications
  - Security: Integrity and confidentiality aspects (to meet security objectives of the TOE)
  - Properties (ex: review, checksum, cypher…)
  - Physical and electrical tranfers are concerned

ADO_IGS.1 - installation, generation and start-up

- **Objectives**
  - to determine whether the procedures and steps for the secure installation, generation, and start-up of the TOE have been documented and result in a secure configuration.
- **Input**
  - Administrator guidance
  - Secure installation, generation, and start-up procedures
  - TOE suitable for testing

ADO_IGS.1 - installation, generation and start-up

- **Installation - Ex: definition of parameters**
- **Generation**
- **Start-up - Ex: ATR, start-up routines...**
- **Description how to verify that all components required for installation have been received (correctness)**
- **Description of Steps for secure IGS**
- **Details on :**
  - Changing security characteristics (of entities under the TSF)
  - Handling exceptions and problems
  - Minimum system requirements for secure installation if applicable

ADV - Development activities

- **ADV_FSP.2 : Evaluation of functional specification**
- **ADV_HLD.2 : Evaluation of High level design**
- **ADV_IMP.1 : Evaluation of implementation representation**
- **ADV_LLD.1 : Evaluation of low-level design**
- **ADV_RCR.1 : Evaluation of Representation correspondence**
- **ADV_SPM.1 : Evaluation of security policy modeling**

ADV_FSP.2 - Evaluation of functional specification

- **Objectives**
  - to determine whether the developer has provided an adequate description of all security functions of the TOE
  - and whether the security functions provided by the TOE are sufficient to satisfy the functional requirements of the ST.

- **Input**
  - ST
  - Functional specification
  - User guidance
  - Administrator guidance

ADV_FSP.2 - Evaluation of functional specification

- **Informal description for EAL4**

- **Description of all the TOE Security function interface**
  - Complete behaviour, protocol
  - Effects, exceptions and error messages

- **Coherence with the user and administrator guidance, ST**
- **TSF fully represented**
- **Argumentation [link TSF - Specifications]**
- **Instantiation of the TOE Functional Requirements**

ADV_HLD.2 - Evaluation of High level design

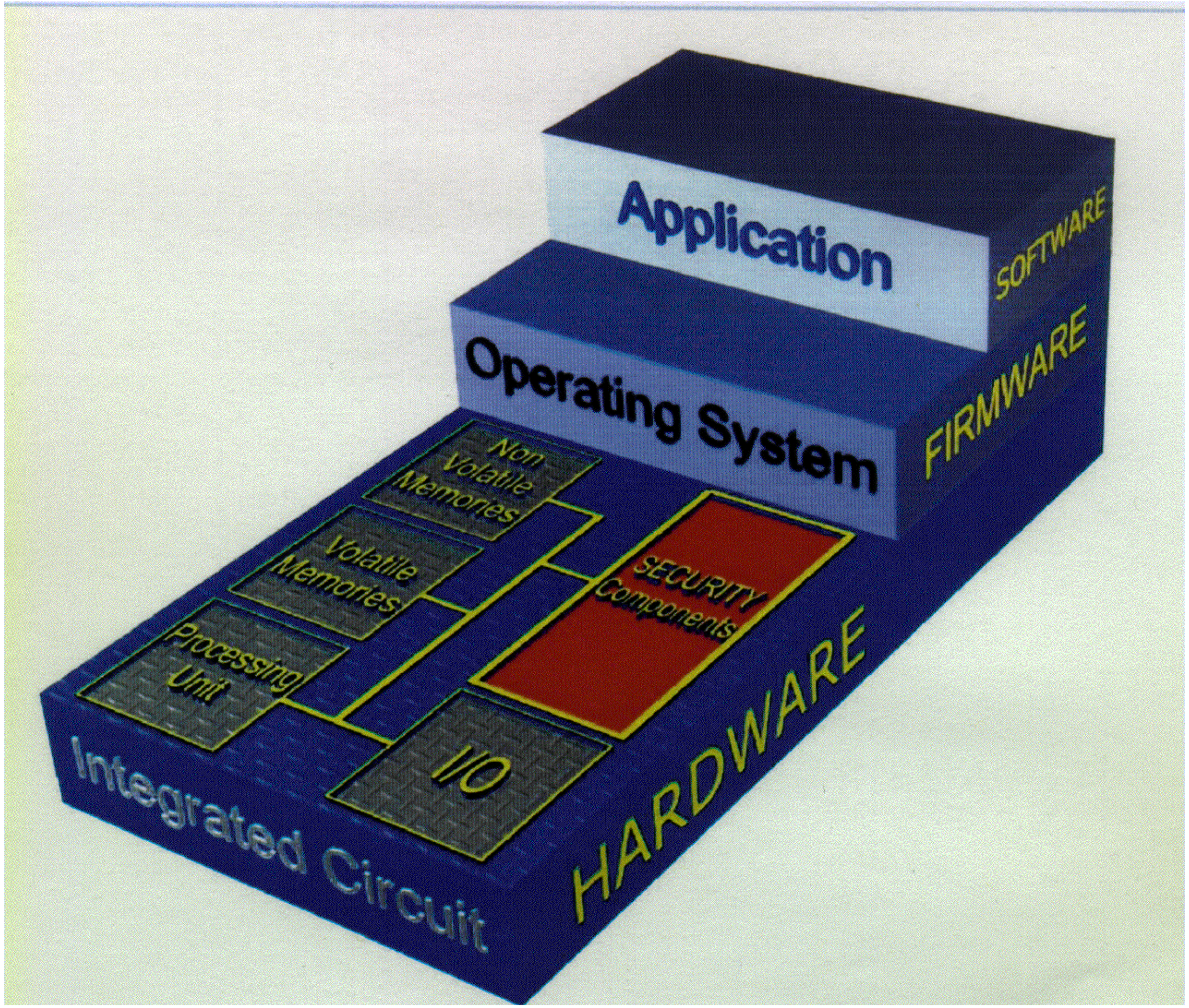
- **Objectives**
  - to determine whether the high-level design is sufficient to satisfy the functional requirements of the ST,
  - to provide a description of the TSF in terms of major structural units with functional coherence,
  - to provide a description of the interfaces to these structural units, and is a correct realisation of the functional specification.

- **Input**
  - ST
  - Functional specification
  - High-level design

Diffusion libre

ADV_HLD.2 - Evaluation of High level design

- .



ADV_HLD.2 - Evaluation of High level design

- **TSF described in terms of subsystems**
- **Description of the functional behaviour of each subsystem**
  - Action, effects in term of security
- **Identification all hardware, firmware, and software required by the TSF (IT environment)**
- **Identification of the interfaces to the TSF subsystems**
  - Purpose, method of use, effects, exceptions, error messages,
  - Identification of interfaces externally visible
- **Description of the separation of the TOE into TSP-enforcing and other subsystems**
- **Accurate and complete instantiation of the TOE SF requirements**

ADV_IMP.1 - implementation representation

- **Objectives**
  - to determine whether the implementation representation is sufficient to satisfy the functional requirements of the ST
  - and is a correct realisation of the low-level design.
- **Input**
  - ST
  - Low-level design
  - Subset of the implementation representation

ADV_IMP.1 - implementation representation

- **Representation is :**
  - Hardware : Schematics, layouts, VHDL
  - Software : source code
- **Unambiguously defines the TSF to a level of detail such that the TSF can be generated without any further design decisions**
  - Compilation of source code
  - Building of hardware from hardware drawings
- **Sufficiently representative, internally consistent**
- **Instantiation of SFR**

⇒ **Training of evaluators, comments**

ADV_LLD.1 - Evaluation of low-level design

- **Objectives**
  - determine whether the low-level design is sufficient to satisfy the functional requirements of the ST,
  - is a correct and effective refinement of the high-level design,
  - and provides sufficient information to support other evaluation activities.
- **Input**
  - ST
  - Functional specification
  - High-level design / low-level design

ADV_LLD.1 - Evaluation of low-level design

- **Description of the TSF in terms of modules**
  - Describes the purpose of each module.
  - Interrelationships between modules (provided security functionality)
  - Dependencies on other modules
- **Description how each of the TSP-enforcing functions is provided**
- **Identification of the interfaces of the TSF modules**
  - Purpose, method of use, effects, exceptions, error messages,
  - Identification of interfaces externally visible
- **Separation of the TOE into TSP-enforcing and other modules**
- **Accurate and complete instantiation of the TOE SFR**

ADV_RCR.1 - Representation correspondence

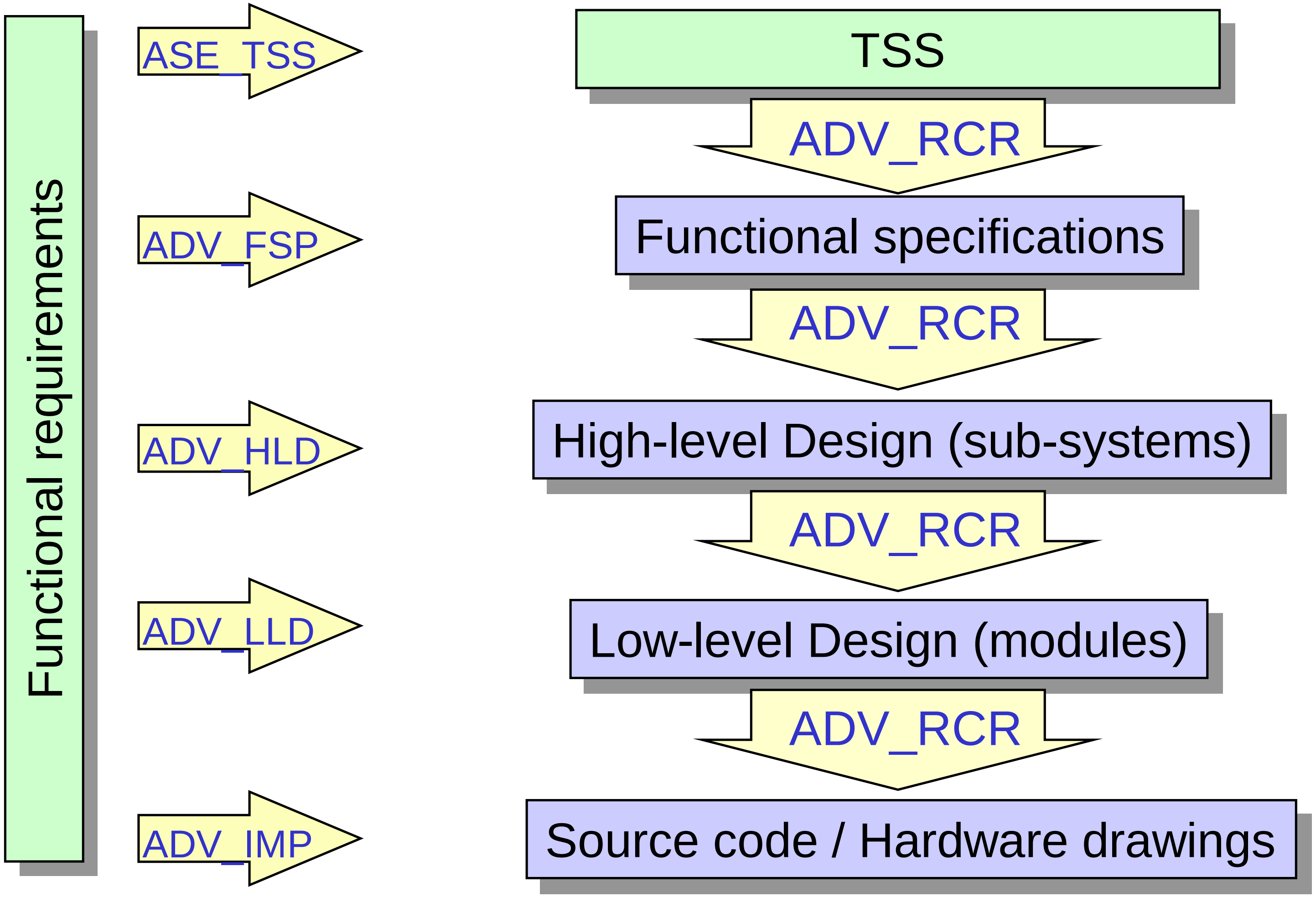
- **Objectives**
  - correct and complete implementation of the requirements of the ST functional specification, high-level design and low-level design in the implementation representation.
- **Input**
  - ST / functional specification / high / low level design
  - Subset of the implementation representation
  - Correspondence TSS - functional specification
  - Correspondence functional specification - high-level design
  - Correspondence high-level design - low-level design
  - Correspondence low-level design - implementation representation

ADV_RCR.1 - Representation correspondence

- Different refinents should be a correct and complete representation of the TOE security functions
- Consistency and accuracy in the correspondence
- Verification of the refinements

ADV_RCR.1 - Representation correspondence

- .


Functional requirements
ASE_TSS
ADV_FSP
ADV_HLD
ADV_LLD
ADV_IMP
TSS
ADV_RCR
Functional specifications
ADV_RCR
High-level Design (sub-systems)
ADV_RCR
Low-level Design (modules)
ADV_RCR
Source code / Hardware drawings

ADV_SPM.1 - Evaluation of security policy modeling

- **Objectives**
  - to determine whether the security policy model clearly and consistently describes the rules and characteristics of the TSP
  - and whether this description corresponds with the description of security functions in the functional specification.
- **Input**
  - ST
  - Functional specification
  - TOE security policy model
  - User, administrator guidance

ADV_SPM.1 - Evaluation of security policy modeling

- **Informal explanatory text**
  - Services or functions available at the external interface
  - See [CEM, §1245]
- **Any TSPs that are explicitly included in the ST are modeled**
- **All security policies in the SFR claimed in the ST are modeled**
- **Rules and characteristics of the model >TOE security behaviour clearly articulated [CEM, §1254]**
- **TSP model rationale**
  - Consistency, completeness with respect to policies described by TSP
- **Functional specification correspondence demonstration of the TSP model**

AGD : Guidance documents activity

- **AGD_ADM.1 : Evaluation of administrator guidance**
- **AGD_USR.1 : Evaluation of user guidance**

AGD_ADM.1 - Administrator guidance

- **Objectives**
  - to determine whether the administrator guidance to system administrative personnel describes how they administer the TOE in a secure manner.

- **Input**
  - ST / functional specification / high level design
  - User, administrator guidance
  - Secure installation, generation, and start-up procedures
  - Life-cycle definition

AGD_ADM.1 - Administrator guidance

- **Administrative SF, interfaces available to the administrator**
  - Method(s) by which the interface is invoked (command-line, programming-language system calls, menu selection...)
  - Parameters to be set (valid and default values)
  - Immediate TSF response, message, or code returned
- **Description how to administer the TOE in a secure manner**
- **Description of all assumptions regarding the user behaviour**
- **Description of all dministrator 's security parameters**
- **Description of each type of security-relevant event relative to the administrative functions that need to be performed**
- **Description of administrative IT environment security requirements**

AGD_USR.1 - User guidance

- **Objectives**
  - to determine whether the user guidance describes the security functions
  - and interfaces provided by the TSF
  - and whether this guidance provides instructions and guidelines for the secure use of the TOE.

- **Input**
  - ST / functional specification / high level design
  - User guidance, administrator guidance
  - Secure installation, generation, and start-up procedures
  - Vulnerability analysis and misuse analysis of the guidance

AGD_USR.1 - User guidance

- **Description of the security functions, interfaces available to users**
  - overview of the security functionality
  - purpose of the security interfaces and functions
- **Description of the use of user-accessible security functions**
- **Warnings about user-accessible functions and privileges (in a secure processing environment)**
- **Responsibilities necessary for secure operation, assumptions**
- **Description of each type of security-relevant event relative to the user functions that need to be performed**
- **Description of user IT environment security requirements**

ALC : Life cycle support activity

- **ALC_DVS.1 : Evaluation of development security**
- **ALC_LCD.1 : Evaluation of life cycle definition**
- **ALC_TAT.1 : Evaluation of tools and techniques**

ALC_DVS.1 - development security

- **Objectives**
  - to determine whether the developer’s security controls on the development environment are adequate to provide the confidentiality and integrity of the TOE design and implementation that is necessary to ensure that secure operation of the TOE is not compromised.

- **Input**
  - ST
  - Development security documentation

⇒ **An audit is scheduled to verify decribed procedures**

ALC_DVS.1 - development security

- **All security measures used in the development environment**
  - Identification of the sites
  - Confidentiality and integrity of the TOE (assets)
  - From design to the implementation
    - *physical, access controls*
    - *procedural*
    - *personnel*
    - *other security measures*
      - the logical protections on any development machines, IT security…
- **Sufficiency of the security measures employed analysed**
- **Documentary evidence (result of procedures application)**

ALC_LCD.1 - life cycle definition

- **Objectives**
  - to determine whether the developer has used a documented model of the TOE life-cycle.
- **Input**
  - ST
  - Life-cycle definition documentation

ALC_LCD.1 - life cycle definition

- **Life-cycle model**
- **Description of the model used to determine that it covers all aspects of the development and maintenance process**
  - Use of the procedures, tools and techniques
  - Minimise the likelihood of security flaws
  - DRC, ERC, Quality and process reviews…
- **Consistent with ACM activities**

ALC_TAT.1 - Evaluation of tools and techniques

- **Objectives**
  - to determine whether the developer has used well-defined development tools (e.g. programming languages or computer-aided design (CAD) systems) that yield consistent and predictable results.
- **Input**
  - Development tool documentation
  - Subset of the implementation representation

ALC_TAT.1 - Evaluation of tools and techniques

- **Development tool documentation > well defined**
- **Unambiguous meaning of all statements of implementation**
  - Programming language specifications (purpose and effect)
    - Schematics symbols
    - VHDL
    - synthesys parameters
    - Assembler (if assembler code in the evaluated TOE)
  - User manuals
  - Caption of symbols…

ATE : Tests Activities

- **ATE_COV.2 : Evaluation of coverage**
- **ATE_DPT.1 : Evaluation of depth**
- **ATE_FUN.1 : Evaluation of functional tests**
- **ATE_IND.2 : Evaluation of independent testing**

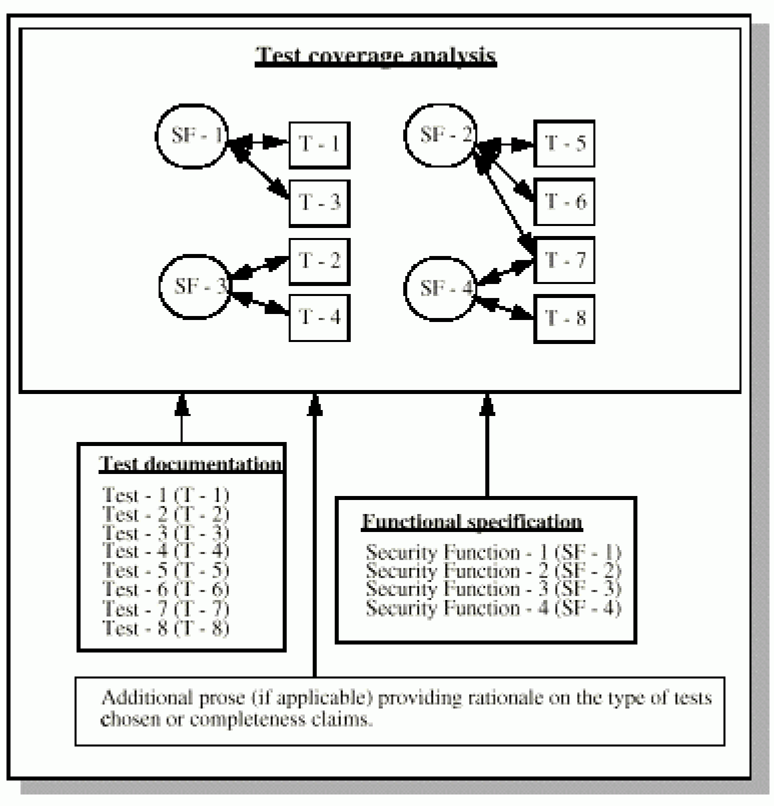
ATE_COV.2 - Evaluation of coverage

- **Objectives**
  - to determine whether the testing (as documented) is sufficient to establish that the TSF has been systematically tested against the functional specification.
- **Input**
  - ST / functional specification
  - High-level design
  - Test documentation
  - Test coverage analysis

ATE_COV.2 - Evaluation of coverage

- **Accurate mapping between the tests identified in the test documentation and the functional specification**
  - Correspondence : table or matrix.
  - Rationale
- **Test plan**
  - Testing approach for each security function of the TSF
  - Suitable to demonstrate the expected behaviour.
- **Test procedures > adequately test each security function.**
- **Correspondence TSF (functional specification) / tests > complete**

ATE_COV.2 - Evaluation of coverage



Test documentation

Test - 1 (T - 1)
Test - 2 (T - 2)
Test - 3 (T - 3)
Test - 4 (T - 4)
Test - 5 (T - 5)
Test - 6 (T - 6)
Test - 7 (T - 7)
Test - 8 (T - 8)

Functional specification

Security Function - 1 (SF - 1)
Security Function - 2 (SF - 2)
Security Function - 3 (SF - 3)
Security Function - 4 (SF - 4)

Additional prose (if applicable) providing rationale on the type of tests chosen or completeness claims.

ATE_DPT.1 - Evaluation of depth

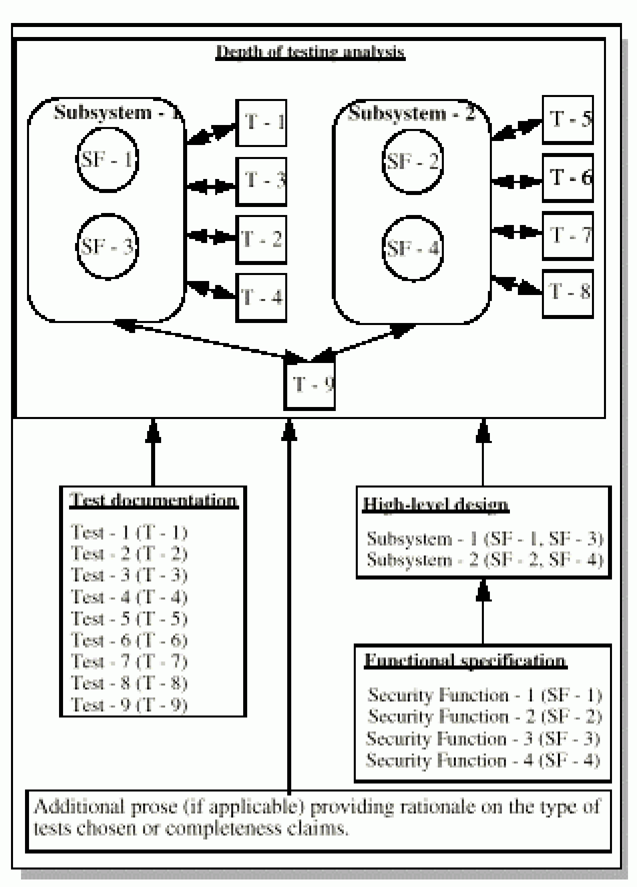
- **Objectives**
  - to determine whether the developer has tested the TSF against its high-level design.
- **Input**
  - ST
  - Functional specification
  - High-level design
  - Test documentation / depth of testing analysis

ATE_DPT.1 - Evaluation of depth

- **Depth : mapping between the tests and high-level design**
- **Testing approach for each SF of the TSF is suitable to demonstrate the expected behaviour**
- **TSF as defined in the high-level design > completely tested**
  - All subsystems
  - All internal interface

ATE_DPT.1 - Evaluation of depth

-



Depth of testing analysis
Subsystem - 1
SF - 1
SF - 3
T - 1
T - 3
T - 2
T - 4
Subsystem - 2
SF - 2
SF - 4
T - 5
T - 6
T - 7
T - 8
T - 9
Test documentation
Test - 1 (T - 1)
Test - 2 (T - 2)
Test - 3 (T - 3)
Test - 4 (T - 4)
Test - 5 (T - 5)
Test - 6 (T - 6)
Test - 7 (T - 7)
Test - 8 (T - 8)
Test - 9 (T - 9)
High-level design
Subsystem - 1 (SF - 1, SF - 3)
Subsystem - 2 (SF - 2, SF - 4)
Functional specification
Security Function - 1 (SF - 1)
Security Function - 2 (SF - 2)
Security Function - 3 (SF - 3)
Security Function - 4 (SF - 4)
Additional prose (if applicable) providing rationale on the type of tests chosen or completeness claims.

ATE_FUN.1 - Evaluation of functional tests

- **Objectives**
  - to determine whether the developer's functional testing demonstrates that all security functions perform as specified.

- **Input**
  - ST
  - Functional specification
  - High-level design
  - Test documentation

ATE_FUN.1 - Evaluation of functional tests

- **Test plans**
  - Identifies the security functions to be tested
  - Describes the goal of the tests performed
  - Test configuration consistent with ST configuration
- **Test procedures**
  - Sufficient instructions establish reproducible
    - test initial conditions including ordering dependencies if any
    - means to stimulate the SFs and to observe the SF’s behaviour
- **Expected test result**
  - Demonstrate the successful execution of the tests
    - concistency with actual results

ATE_IND.2 - Evaluation of independent testing

- **Objectives**
  - to determine whether the TOE behaves as specified and to gain confidence in the developer's test results by independently testing a subset of the TSF and by performing a sample of the developer's tests.

- **Input**
  - ST / Functional specification
  - User & administrator guidance
  - Secure installation, generation, and start-up procedures
  - Test doc. / test coverage analysis / depth of testing analysis
  - TOE suitable for testing

ATE_IND.2 - Evaluation of independent testing

- **Test configuration is consistent with the configuration under evaluation as specified in the ST**
- **TOE installed properly and in a known state**
- **Set of resources provided by the developer**
- **Evaluator’s tests**
  - test identical
  - new tests
  - verification of the SOF

AVA : Vulnerability assessment activity

- **AVA_MSU.2 :** Evaluation of misuse
- **AVA_SOF.1 :** Evaluation of strength of TOE security functions
- **AVA_VLA.2 :** Evaluation of vulnerability analysis

AVA_MSU.2 - Evaluation of Misuse

- **Objectives**
  - to determine whether misleading, unreasonable and conflicting guidance is absent from the guidance,
  - whether secure procedures for all modes of operation have been addressed, and whether use of the guidance will facilitate detection of insecure TOE states.

- **Input**
  - ST / functional specification /high-level design / low-level design
  - Subset of the implementation representation
  - TOE security policy model, user, administrator guidance
  - Secure installation, generation, and start-up procedures

AVA_MSU.2 - Evaluation of Misuse

- **guidance clear, complete and internally consistent**
  - Identifies all possible modes of operation of the TOE, error
    - operation following failure or operational error,
    - consequences and implication for maintaining secure operation
- **All assumptions about the intended environment articulated**
- **Mappings from design specification, in particular the functional specification, to the guidance > complete**
- **TOE configured and used securely using only supplied guidance**
  - Detectection of insecure states by users, administrators
  - Guidance provided for secure operation in all modes of operation of the TOE

AVA_SOF.1 - Strength of TOE SFs

- **Objectives**
  - to determine whether SOF claims are made in the ST for all probabilistic or permutational mechanisms
  - and whether the developer’s SOF claims made in the ST are supported by an analysis that is correct.

- **Input**
  - ST / functional specification
  - High-level design / the low-level design
  - Strength of function claims analysis

AVA_SOF.1 - Strength of TOE SFs

- SOF claims analysis for each security mechanism for which there is a SOF claim expressed as a SOF rating
- Claims analysis > each SOF claim met or exceeded
- Correctness of assumptions supporting the analysis
- Correctness of algorithms, principles, properties and calculations supporting the analysis
- No probabilistic or permutational mechanisms without a SOF claim

AVA_VLA.2 - Vulnerability analysis

- **Objectives**
  - To determine whether the TOE, in its intended environment, has vulnerabilities exploitable by attackers possessing low attack potential.
  - For EAL4+, PP/9806, AVA_VLA.4 is chosen to reach high attack potential.
- **Input**
  - All the documentation concerning the TOE (except test documentation)

AVA_VLA.2 - Vulnerability analysis

- **vulnerability analysis**
  - Vulnerabilities in at least all evaluation deliverables
  - Public domain information
  - Rationale > why it is not exploitable
  - Consistent with the ST and the guidance

- **Developpers penetration tests**
  - Penetration testing effort
  - Outlining the testing approach
  - Configuration
  - Depth and results

AVA_VLA.2 - Vulnerability analysis

- **Evaluators penetration test**
  - Conducted at the end of the evaluation
  - Search for additional security vulnerabilities

- **Resistant to an attacker possessing a High attack potential ?**
  - All exploitable vulnerabilities
  - And “residual vulnerabilities”