D3.1 –Analysis of Existing Assurance Methodologies and Tools

Security Assurance Framework for Networked Vehicular Technology

Abstract

SAFERtec proposes a flexible and efficient assurance framework for security and trustworthiness of Connected Vehicles and Vehicle-to-X (V2X) communications aiming at improving the cyber-physical security ecosystem of "connected vehicles" in Europe. The project will deliver innovative techniques, development methods and testing models for efficient assurance of security, safety and data privacy of ICT related to Connected Vehicles and V2X systems, with increased connectivity of automotive ICT systems, consumer electronics technologies and telematics, services and integration with 3rd party components and applications. The cornerstone of SAFERtec is to make assurance of security, safety and privacy aspects for Connected Vehicles, measurable, visible and controllable by stakeholders and thus enhancing confidence and trust in Connected Vehicles.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

DX.X & Title:	D3.1 Assurance Methodologies and Tools
Work package:	WP3 Assurance Framework
Task:	T3.1 Assurance Methodologies and Tools
Due Date:	M12
Dissemination Level:	PU
Deliverable Type:	R

Authoring and review process information	
EDITOR Sammy HADDAD / OPP	**DATE** 19/07/2018
CONTRIBUTORS Sammy HADDAD / OPP Costas Lambrinoudakis / UPRC Kostas Maliatsos / UPRC Panagiotis Pantazopoulos / ICCS	**DATE** 14/05/2018 - 10/9/2018
REVIEWED BY Sylvia Capato / SWA Panagiotis Pantazopoulos / ICCS	**DATE** 13/06/2018 – 09/10/2018
LEGAL & ETHICAL ISSUES COMMITTEE REVIEW REQUIRED?	
NO	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Document/Revision history

Version	Date	Partner	Description
V0.1	02/02/2018	OPP	First draft of assurance frameworks state of the art and assurance framework
V0.2	20/04/2018	OPP	Framework update after UPRC and ICCS reviews and comments
V0.3	03/05/2018	OPP	Update of assurance tools with UPRC inputs, assurance level and assurance continuity sections
V0.4	13/06/2018	SWA	Peer review report submitted
V0.8	19/07/2018	OPP	Revised
V0.9	09/10/2018	ICCS	Comments and corrections
V1.0	23/10/2018	OPP	First complete version

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Table of Contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Table of Figures

List of Tables

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Acronyms and abbreviations

Abbreviation	Description
ANSSI	Agence National de Sécurité des Systèmes d'Information
CAM	Cooperative Awareness Message
CC	Common Criteria
CCMS	Cooperative ITS Credentials Management System
CSP	Critical Security Parameters
CSPN	"Certification de premier niveau"
C-ITS	Cooperative ITS
DENM	Decentralized Environmental Notification Message
IT	Information Technology
ITSEF	Information Technology Security Evaluation Facility
OBU	On-Board Unit
OS	Operating System
OSP	Organizational Security Policy
PP	Protection Profile
SFR	Security Functional Requirements
ST	Security Target
TOE	Target Of Evaluation
TSF	TOE Security Functions
V2I	Vehicle to infrastructure
V2X	Vehicle to everything

Table 1: List of Abbreviations

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Executive Summary

This document presents the current state of the art on security, privacy and safety assurance methodologies and tools.

The first section of the document presents existing assurance and general security evaluation approaches.

The second part of the document presents the state of the art for assurance enhancing methods, models, implementation tools and testing methods dedicated to providing trust in product and system security. In this section we will also discuss how these tools can be integrated in existing assurance methodologies' activities.

Finally, we introduce the (preliminary) design of the SAFERtec assurance framework together with its associated evaluation tasks and tools. We discuss how it enhances existing methodologies and how it suits better the ITS requirements than other approaches.

The proposed SAFERtec framework highlighted herein, will be further enhanced (and modified, if need be) as the actual WP3 technical work progresses; all newest achievements will be reported in the upcoming deliverables.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

1 Introduction

This deliverable introduces the SAFERtec assurance framework. The main goal of SAFERtec is to propose a flexible and efficient assurance framework for security and trustworthiness of Connected Vehicles and Vehicle-to-Infrastructure (V2I) communications aiming at improving the cyber-physical security ecosystem of "connected vehicles" in Europe.

Achieving security assurance is a very challenging task. Security assurance is mainly about establishing trust in the fact that a product fulfils its security requirements. There is no known fast or easy way to assure that a product is secure. Accordingly, it is very difficult to present convincing arguments that a specific assurance method and its associated tools are the most appropriate ones. Evaluating security has been an active research, industrial and governmental topic for the last 3 decades. Many approaches have been proposed but very few of them have enjoyed a global consensus. Actually, only one managed to get an international recognition, regardless the criticism it faces. The most widely recognized one, is the *Common Criteria for Information Technology Security Evaluation (CC)* standardized in *ISO/IEC 15408 Information technology – Security techniques – Evaluation criteria for IT security.*

This global evaluation approach that remains agnostic to any specific technology is both the most recognized and criticized one, the second assertion being directly related to the first one. If the two ends of the security assurance spectrum can be easily understood and accepted, i.e.:

- *the lowest end* (of assurance) - doing nothing to prove the security providing of course no specific confidence at all, expecting "security by chance" and
- *the highest end*- formally prove that every possible execution of the system satisfies its intended behaviour;

the real challenging assurance assessment part lies in-between. There, we have most of the cases of IT products providing security functions.

In this deliverable, we will detail the state of the art of existing assurance and security evaluation methods. We will discuss the pros and cons of the different approaches and try to identify the inherent limitations of these processes.

Then, we will study existing tools and methodologies related (or being able) to provide security assurance. In other words, we will examine tools or processes that can provide confidence on the fact that the security requirements defined for a specific system or product (also called Target Of Evaluation (TOE)) are satisfied. Those tools or methodologies can take many forms and address many different aspects of the TOE's life cycle. For instance, they can study:

- the TOE development environment or processes in order to guarantee the quality of the TOE,
- the test suit employed by the *developer*[1] or by an independent body to verify or validate its actual behavior,
- verify the use of code analyser or automated API tester to validate some product properties or conformity (to certain pre-defined requirements),
- the method and tools used to specify the product and its functions, operational metrics to observe real TOE behaviour in its operational environment.

[1] Throught the document we use the term 'developer' to refer to the responsible party/individual for the development of the corresponding software.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

These observations provide elements of proof that the TOE fulfils the requirements that have been earlier formalized.

From the aforementioned analysis we will both define and discuss the new SAFERtec assurance framework dedicated to ITS systems. We will define a generic framework, i.e.: evaluation tasks and associated actors; and identify tools that will be developed during the project in order to provide assurance in the most efficient way.

1.1 Purpose of the Document

The purpose of the document is to study the state of the art of IT assurance methods and tools, and introduce an efficient approach dedicated to ITS systems.

1.2 Intended readership

Public. (any interested person/party)

1.3 Inputs from other projects

The French IRT SystemX project ISE[2], proposed a dedicated assurance framework for ITS product evaluation, CARSEM. Our work re-uses some of the concepts of that proposal. Importantly, SAFERtec enhances the above proposal by introducing more precise evaluation methods and tools to run this general assurance framework.

1.4 Relationship with other SAFERtec deliverables

This deliverable receives as input all the functional security, privacy, safety objectives and requirements identified for ITS systems and particularly for the 'connected vehicle system' as explored in the context of the SAFERtec WP2.

[2] https://www.irt-systemx.fr/en/project/ise/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

2 Global state of the art for assurance frameworks

Several IT evaluation schemes exist. Their main objective is to validate the security functions of a product or system. A subset of them ends with an official certificate delivered by a certification authority. In all cases there is an evaluation process to be done and the level of recognition then depends on the level of maturity of the method and the one who runs it. In the case of a certification, it is the certificate that determines the level of the achievable recognition. SAFERtec does not aim at defining a certification process (i.e. no need for an official certification body) but only a faster evaluation process that does not have to go through the burden of an administrative task associated with certification. However, in this section we present the state of the art for both evaluation and certification processes.

There is a difference in evaluating or certifying a product and approve an entire system. In this document and more generally in the SAFERtec project, we aim at studying mainly the evaluation of the most important ITS system i.e., the OBU which holds a central position in the entire ITS future system. We do not want to approve the whole ITS system at once but only certify part of it and validate the general security architecture to provide enough assurance in the complete system security.

Thus, in this section we present the existing evaluation processes and schemes for IT products.

But first we start by describing general challenges for security assurance.

2.1 Security evaluation methodologies generalities and common challenges

Before comparing different evaluation schemes and methodologies, we start by identifying the main and most important aspects that make the difference between existing security evaluations.

All existing IT security evaluation methods address the following three directions:

- What must be evaluated?
  - Which product and which version of the product?
  - Which function of the product?
  - In which environment and for which type of threat?
- Which evaluation activities?
  - Evaluate the development
  - Evaluate the product architecture
  - Test the external/internal interfaces
  - Analyze the code, the guides, etc.
- Who is competent and must be in charge of what:
  - Who is the evaluation authority in charge of defining and managing the evaluation activities to guarantee the overall evaluations expectations?
  - Who will pay and be the sponsor of the evaluation?
  - Who has the expertise and required test environment?
  - What does the developer have and what information must he provide for the evaluation of its product?
  - What is the end user's point of view?

The above three dimensions correspond to what is generally called:

- The Security Target (ST)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- The assurance components
- The evaluation scheme.

All IT security evaluation schemes have their own interpretation of what is important for these three dimensions and how to obtain them.

It is important to understand that there is no universal solution for the problem of IT security evaluation and all known solutions are criticized. In fact, they all have different advantages and drawbacks.

Security evaluation is a difficult problem and will probably remain so for a long time because IT systems are complex and they evolve very rapidly. Whether it is feasible or not to obtain a formal proof of systems security, the current state of the IT technologies makes a relevant effort unworthy (mainly because it is too expensive).

ITS systems are directly concerned by this observation. These systems are relatively new, so they do not benefit from the years of real security experience, and they are complex (system of systems, large applications, etc.). So, it will not be an easy task to define and adopt a universally recognized evaluation scheme for ITS products and systems.

Four main evaluation approaches exist so far to tackle this general challenge. We discuss them briefly here.

2.1.1 Conformity Checks

Conformity Checks (also called compliance assessment) is a form of evaluation that validates a product or system compliance to a specific reference. This approach needs to have a reference conformity list. This list has to be kept up to date and has to be relevant for the product type and its real needs in terms of functionality and security. There are two main limitations to the conformity check approach. First, the definition and maintenance of relevant conformity lists can be difficult or even infeasible in an industrial context (i.e., too many updates needed, no agreement on the conformity requirements, scope of conformance too restrictive, etc.). Also, anything not conformant to (a part of) the conformity list cannot be validated. On the other side, conformity checks provide usually the fastest and cheapest evaluation scheme compared to other methods, providing comparable levels of confidence. Also, the evaluation results are simpler to understand and easily comparable since every test is known in advance and they are the same for every product evaluated.

A main certification (and thus evaluation) scheme that defines a normalized test suite suitable for Conformity Checks is the FIPS 140-2 standard [1]. This certification only concerns cryptographic products. The FIPS are public standards developed by the United States federal government, aiming at ensuring some computer security and interoperability for the US governmental Information Systems.

Contrary to other frameworks, such as ITSEC, CC or the French CSPN [2] [3] [4] [5] [6] FIPS evaluations do not need the specification of a security target. The list of functions and tests to be done is directly defined by the FIPS 140-2 standard, which indirectly defines the security target together with the assurance component through the list of conformity checks.

In this approach, since the test requirements are defined in the standard, they age with it and the standard has to be rewritten every time new security paradigms are required (i.e., new threats, new needs, etc.). For this reason, the FIPS 140-2 standard foresees to be reviewed every five years, whereas such a standard in the ITS world should be typically reviewed every 6 months considering the rapid evolution of the system. Also, even if cryptographic functions are quite well recognized and very limited in complexity and numbers, this is not the

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

case when we consider the full implementation of an ITS architecture. Such architecture includes OSs, communication and security stacks, sensors, applications and so on. Cryptographic functions are a very limited subset of those systems and scaling the methodology would be at least as expensive as developing the system themselves.

In many industrial sectors and when feasible, this scheme is the preferred one – see for example the Compliance Assessment process specified by the C-ITS Platform[3], the US Certification program for Connected Vehicles and ETSI ITS validation platform for standardized protocols. But such an approach can only partially cover ITS security validation and so far, nothing close to the beginning of a recognized and validated set of security requirements and their associated tests exists (despite the fact that this approach is regularly promoted).

2.1.2 Vulnerability tests

This approach simply defines an evaluation perimeter, not necessarily forming a real complete ST. Usually it only defines the product, the tests environment and associated limitations. Then an expert runs any tests of his/her choice during a predefined time on the defined scope. At the end, the result is the set of potential vulnerabilities identified by the tester. If no vulnerability is found, then the evaluation result states that the product resisted to an attacker during a number of days equals the evaluation time.

Thus, this method allows validating the product's security level, providing low to medium assurance level. Also, on average the results are obtained faster than other methodologies; note that common tests take 20 to 30 days.

The problem with this methodology is that there is a great need of confidence in the tester competences. Also, results are not fully consistent or directly comparable since two testers are free to use completely different tests for the very same product.

A formalized approach falling under this category is the French CSPN (cf. section 2.3.1) where a detailed ST is required and the number of vulnerability test days is predefined, 25 days for every product. This process is the only one that provides a certificate signed by the prime minister and recognized nationally.

2.1.3 Assurance framework

The Assurance framework approach is the more complete and exhaustive approach. It provides the highest assurance levels (i.e., level of confidence in the product security), but it is generally more expensive and time consuming. It also requires the involvement of rare and expensive accredited evaluators.

The CC is inspired from two important assurance schemes appeared in United States and Europe: [7] and [2].

The first version of the Common Criteria for Information Technology Security Evaluation, known as Common Criteria (CC) dates back to 1994 and the last version to be standardized [4] was released in 2009. Since then regular revisions have been done but the global approach has not change. The current version accessible on the common criteria portal and used for evaluations is the 3.1 Release 5.

It keeps the main concepts of ITSEC: (i) the notion of the need of a proper ST target, (ii) the decomposition of the evaluation in generic evaluation tasks independent of any product or security requirements, (iii) the definition of

[3] https://ec.europa.eu/transport/sites/transport/files/2017-09-c-its-platform-final-report.pdf

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

several evaluation assurance levels, each providing a set of more stringent evaluation tasks and evidences requirements.

Eventually the CC provides a complete description and a reference set of security requirements to write formalized STs and the most extensive list of evaluation activities including any activities empirically recognized as having a potential impact on the final product security.

The CC global approach consists of the evaluation of every product life cycle elements that helps to demonstrate that security requirements identified in the ST can be traced to the real product delivered to the end user. It proposes to evaluate the product life cycle management, the product architecture and full specification, the guides provided with the product to demonstrate that it can be easily used with the proper security configuration, the functional test run on the product and finally the vulnerability test to complete the whole assessment that the product fulfills the requirements stated in the ST and that those requirements cannot be bypassed. Vulnerability tests and conformity checks are included in the CC and are only subparts of a complete CC evaluation. No other methodology covers as many aspects or is as well structured. That is why it is the best approach and accordingly the most expensive one. Also, it is the only one to benefit from an official international recognition agreement.

2.1.4 Security metrics and other evaluation approaches

The three aforementioned approaches are the most commonly used ones. However, over the last three decades many researchers and practitioners have addressed the general problem of IT products validation, to try to find more specific and formalized approaches. So far, no fully satisfying (i.e. universal recognition with no cons) solution has been found (and it will probably never be).

A comprehensive overview of the various efforts made on the evaluation and measurement of IT security domain 10 years ago can be found in [8] and [9]. It covers software, standards ([10], [11]), taxonomies ([12], [13]), metric definitions ([14], [9]), methodologies ([15], [16]), security databases ([13]), etc.

However, many of them face the criticism of security evaluation challenges ([17], [18]), relying on sole security expert's knowledge or being not adapted to real dynamic systems. And even if works are still on going and efforts are made to try to enhance evaluation methodologies, there is no new proposed solution and the same three main (aforementioned) approaches are used.

2.2 Security Assurance frameworks

There are two main types of security validation, evaluation or accreditation processes, those made for products and those for systems. The more formal and structured one are for products: CC, ITSEC, TCSEC; when system security assessment includes more generic definitions of procedures, it is also called Information Security Management System (ISMS) such as [19], [20].

In fact, the main problem in security assurance frameworks comes from the fact that assessing security properties of an IT product fully depends on the product itself: its purposes, the technologies used to implement it, its functional and security architecture, its operational environment (e.g. users, interconnections), etc; and finally, the current state of the art of attacks.

All these parameters cannot be constantly standardized for every possible IT product in an up-to-date manner.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Clearly, we cannot evaluate in the same way products such as: a firewall, a data base, a web site or an operating system. It is also very hard to compare the results of any evaluation of this product, since even if they would belong to the same categories they would still be different and not subject to the same sets of attacks and threats; that is because of the technologies used or their operational environment.

Thus, every credible evaluation framework takes this observation as an axiom and does not try to provide a methodology to assess overall security rating, since there is no such universal security scale. All known methodologies adopt the same general structure:

1. Identify the product to be evaluated
2. Define the security problem
    1. Identify the assets to be protected
    2. Identify the threats for the assets to be mitigated
3. Defining the security functions to be validated for that product to mitigate the identified threats
4. Defining a set of evaluation task to apply for the validation of the product's security functions (possibly set of tasks dedicated to the specific product type or category)
5. Defining specific tests for the product to be evaluated

A main difference between the methodologies relies in:

- either each of these points are directly defined by the methodology and thus directly constrained by it (limiting the possible application of the methodology);
- or the methodology asks for these points to be defined, leaving it more flexible.

Another main difference in the referenced approaches is the fact that the scope of evaluation (functionality evaluated) and the assurance level (evaluation tasks to validate the functions to be evaluated) may be independent from each other or not.

2.2.1 Orange book

The first widely used assurance scheme was the Trusted Computer System Evaluation Criteria [7] , commonly known as the TCSEC or "Orange Book".

The evaluation methodology is limited to operating systems. Originally published in 1983 and later updated in 1985, it was used by the US Department of Defense (DoD) in an evaluation scheme operated by the National Computer Security Center (NCSC). The TCSEC criteria are directly intended to match the security policy and requirement of the US DoD. The TCSEC was officially cancelled in 2002 and then replaced by the Common Criteria in 2005.

The TCSEC defines seven sets of evaluation criteria called classes (D, C1, C2, B1, B2, B3 and A1), grouped into four levels:

- D – Minimal protection,
- C – Discretionary protection,
- B – Mandatory protection
- and A – Verified protection.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Each criteria class covers four aspects of evaluation: Security Policy, Accountability, Assurance and Documentation. The criteria for these four areas become more detailed from class to class and form a hierarchy (whereby D is the lowest and A1 the highest). Each class covers both functionality and confidence requirements meaning that it defines both the expected security functions to be found in the system and the verification to do on these functions.

Thus, this methodology mixed all the different aspects of evaluation, the security target, the assurance components and evaluation tasks into predefined evaluations sets. It cannot be used for products that do not provide all the expected functions at the intended level of evaluation.

Security functions are only evaluated if the chosen level of evaluation imposes it. Thus, even if extra security functions are implemented, they will not be evaluated. The developer can't choose (either add or remove) which security function should be evaluated.

2.2.2 The ITSEC approach

The ITSEC was first published in May 1990 in France, Germany, the Netherlands, and the United Kingdom based on existing work in their respective countries. Since the launch of the ITSEC in 1990, a number of other European countries have agreed to recognize the validity of ITSEC evaluations.

The ITSEC was the first general framework for IT security product certification. The ITSEC is a document that describes a structured set of criteria for evaluating security.

It is the first evaluation scheme to introduce the need of evaluation based on specific security target. It defines the mandatory sections to be included in the ST and a description of what is expected in each of these sections:

- Either a System Security Policy or a Product Rationale.
- A specification of the required security enforcing functions.
- A definition of required security mechanisms (optional).
- The claimed rating of the minimum strength of mechanisms.
- The target evaluation level.

The ITSEC provided a classification of security functions to be used to write ST required security enforcing functions:

- Identification and Authentication
- Access Control
- Accountability
- Audit
- Object Reuse
- Accuracy
- Reliability of Service
- Data Exchange

Each of these classes is generally defined in ITSEC and it does not prescribe the use of particular proprietary or standardized methods or styles for the specification of security functions.

ITSEC introduces the notion of security function efficiency that did not exist in the Orange book. In fact, it identifies two evaluation aspects: effectiveness and correctness.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

The efficiency is the study and the evaluation of the capability of the security function to resist to a certain level of attack. This is what the "claimed rating of the minimum strength" section in the ST is used for. It is required to specify which of the three attack levels the TOE must be able to resist to, i.e.:

- Basic: it shall provide protection against random accidental subversion, although it may be capable of being defeated by knowledgeable attackers.
- Medium: it shall provide protection against attackers with limited opportunities or resources.
- High: it should only be defeated by attackers possessing a high level of expertise, opportunity and resources, successful attack being judged to be beyond normal practicality.

This is directly tested via vulnerability tests but also it requires specific documentation to be provided by the developer to present evidence that the structure and conception of the product should guarantee its capacity to resist.

Also, it is the first to introduce an assurance level, independent of the security functions to be evaluated. The assurance levels are based on different levels of requirements for evidences for the developer, classified in what the ITSEC calls evaluation phases:

- Requirements: corresponding to the evaluation of the ST
- Architectural Design: corresponding to the verification of high level definition and design of the TOE
- Detailed Design: corresponding to the evaluation of the correspondence of the architectural design evidences to more detailed software and hardware evidences
- Implementation: corresponding to the evaluation of the TOE implementation

For each of these phases specific evaluation requirements are made in each of the 7 evaluation levels.

Thus, the ITSEC is the first general evaluation framework to propose to evaluate products for specific STs and defining generic evaluation tasks structured in different phases. It was thus the first evaluation framework to be used internationally to evaluate a wide range of product types, regarding generic but comparable common requirements sets.

2.2.3 The Common Criteria for Information Technology Security Evaluation (CC)

The Common Criteria for Information Technology Security Evaluation, commonly named more simply Common Criteria (CC), is an internationally used evaluation framework. It is defined and maintained by an international community. The latest version of the documents defining the CC together with other documents defining the level of international recognition, supporting documents for the methodology application on specific cases or the list of certified product or testing laboratories can be found on the common criteria portal (www.commoncriteriaportal.org).

The latest version of the CC available on the portal is the version 3.1 revision 5 which is different from the last standardized version. The last ISO standardized version is the version 2.3 as ISO/IEC 15408-1:2009 Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 1: Introduction and general model. The different evaluation schemes use generally the last 3.1 revision 4 version.

The first version of the CC dates back to 1994. The last version to be standardized was in 2009. The first revision of version 3.1 dates from 2006. Since then only minor revisions have been done and the document has stabilized.

CC is directly inspired from the previous assurance evaluation initiatives: TCSEC, ITSEC and the Canadian adaptation of the TCSEC the CTCPEC.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

It keeps the main concepts developed in the ITSEC:

1 The notion of the need of a proper ST target being the “what must be” and “what has been” evaluated.
2 The decomposition of the evaluation in tasks independent of any product specificities.
3 The definition of several evaluation assurance levels, each providing a set of more stringent evaluation task and evidences requirements.

The CC are decomposed in three parts each corresponding to one document:

- Part 1: Introduction and general model
- Part 2: Security functional requirements
- Part 3: Security assurance requirements

The first part is an introductory document that defines all the CC vocabulary and the different roles and interest for the different participant of an evaluation.

The most important concepts defined or redefined by the CC are:

- The Target of the Evaluation (TOE): the product or the system to be evaluated.
- The Security Target (ST): the document specifying TOE and the evaluation tasks.
- Protection Profiles (PP): Generic ST defining only evaluation tasks for a generic type of product.
- The Security Functional Requirements (SFR): the specification of the security functions that the TOE must implement.
- The TOE Security Functionality (TSF): the part of the TOE where the SFR are implemented.
- The TSF Interfaces (TSFI): the interfaces used by the users to interact with the TSF.

Also, this document defines the different actors and their roles in the evaluation.

The second part presents a standardized common set of Security Functional Requirements (SFR), i.e. a formalization of the most common security function, e.g.:

- Security audit data generation
- Non-repudiation of origin
- Cryptographic key management
- Access control policy
- Information flow control policy
- Rollback
- User authentication
- Anonymity
- Fail secure

As for the ITSEC, those security functions are presented and classified within 11 classes:

- CLASS FAU: SECURITY AUDIT
- CLASS FCO: COMMUNICATION
- CLASS FCS: CRYPTOGRAPHIC SUPPORT
- CLASS FDP: USER DATA PROTECTION
- CLASS FIA: IDENTIFICATION AND AUTHENTICATION
- CLASS FMT: SECURITY MANAGEMENT
- CLASS FPR: PRIVACY
- CLASS FPT: PROTECTION OF THE TSF

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 732319

- CLASS FRU: RESOURCE UTILISATION
- CLASS FTP: TRUSTED PATH/CHANNELS

The part 2 also defines how to structure and write one of the most important documents of an evaluation, the Security Target (ST). This critical document will define what is the product and in which precise version has to be or had been evaluated and for which function. The particularity of the CC is that all the evaluation process pertains to providing proofs to validate the SFR in the product. To do that, all the documents provided by the developer will have to trace the correct implementation of the SFR at the different level of the product life cycle and conception. Thus, all or most of the documents provided must clearly identify this traceability and thus make references to these SFR. This is one of the reasons why, evidences provided for the evaluation are dedicated to the evaluation and are usually not the regular documentation (product specifications, product architecture, user guides, etc.) produced by the developer.

Finally, the third and last part of the CC presents the evaluation tasks to be done to evaluate the product. The tasks are presented in this general way: description of the goal of the task, its dependencies with other evaluation tasks, evidence requirements for the developer, evaluation activities to be done by the evaluator. Different levels are presented for each task. At the end of the document they are combined to form seven Evaluation Assurance Level (EAL), EAL 1 to EAL7, each of them increasing the level of requirements and verification to be done on the TOE and evidences provided by the developer.

CC defines 8 assurance classes, decomposed each in several assurance families.

Each assurance class is assigned a unique name. The name indicates the topics covered by the assurance class.

A unique short form of the assurance class name is also provided. The convention adopted is an “A” followed by two letters related to the class name. Assurance classes are then decomposed in families. A unique short form of the assurance family name is also provided. This is the primary means used to reference the assurance family. The convention adopted is that the short form of the class name is used, followed by an underscore, and then three letters related to the family name.

Here is an overview of this classes and families:

- PROTECTION PROFILE EVALUATION (APE)
- SECURITY TARGET EVALUATION (ASE)
- LIFE-CYCLE SUPPORT (ALC)
  - Life-cycle definition (ALC_LCD)
  - Development security (ALC_DVS)
  - Configuration Management capabilities (ALC_CMC)
  - Configuration Management scope (ALC_CMS)
  - Delivery (ALC_DEL)
  - Flaw remediation (ALC_FLR)
  - Tools and techniques (ALC_TAT)
- DEVELOPMENT (ADV)
  - Security Architecture (ADV_ARC)
  - Functional specification (ADV_FSP)
  - TOE design (ADV_TDS)

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 732319

  - Security policy modelling (ADV_SPM)
  - Implementation representation (ADV_IMP)
  - TSF internals (ADV_INT)
- GUIDANCE DOCUMENTS (AGD)
  - Preparative procedures (AGD_PRE)
  - Operational user guidance (AGD_OPE)
- TESTS (ATE)
  - Functional tests (ATE_FUN)
  - Coverage (ATE_COV)
  - Independent testing (ATE_IND)
- VULNERABILITY ASSESSMENT (AVA)
- COMPOSITION (ACO)

Two of these classes are not used in common evaluations.

The APE class only concerns the evaluation of what the common criteria call protection profiles (PP). PPs are often used in IT security to define generic security requirement for a family of product (e.g. firewall, cryptographic modules, Network Intrusion Prevention System (NIPS), Trusted Signature Creation Module, etc.).

CC defines a very specific and formalized format to write CC-conformant PPs. These PPs are more precise and use CC language to specify what the CC call security functional requirements for a family of product. These PPs can then be used as a reference for what has to be evaluated in these products. Then for each evaluation of such type of product an ST can be written claiming conformance to this PP.

As these PPs have to be written and to be conformant to the CC format described in the standard, CC also provides the description of the assurance classes to evaluate if a PP is in fact conform to the CC format. The use of the class is thus completely independent from the product evaluation.

The second assurance class that is generally not (and to our knowledge has never been) used, is the ACO class. This class has been defined to provide a solution to the composition of assurance. In fact, product evaluation always faces the problem that a certified product is never used alone but is rather integrated in a wider system. As such, even if the product is correctly evaluated the way it interacts with its environment always has an impact on its security properties. Confidence in a product should not therefore, imply confidence in its integration with other system elements, even if these elements are also certified.

To overcome this problem, CC defined a way to compose the confidence in certified products when they are used in a combined way. This assurance class does not provide a way to get overall certification but rather another level of confidence that the composition preserves independent certified product properties.

Also, not all the aforementioned assurance class and family are used for all EAL.

The goal of each activity is described in slightly more details the following sections.

2.2.3.1 Security target evaluation (ASE)

One of the main elements of any relevant security or assurance evaluation is the ST. This evaluation task has to be done before any other since it is the starting point and specification of the evaluation objectives. That

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

document must contain the set of *Security Functional Requirement* (SFR) that has to be validated. It defines what has to be evaluated. This is the objective of the evaluation.

2.2.3.2 Life-cycle support (ALC)

The second element in the evaluation process of the CC is the evaluation of the development environment and control process used during the life cycle of the product.

This assurance component has one major goal, which is to ensure to the user the integrity of his product. To do that the evaluator evaluates the life-cycle management of a product, i.e. development plan describing how specification, conception, coding, tests, etc. are handled by the developer, error and product modification are fully managed and so that unwanted code modification cannot be done. It evaluates the physical and IT security of the development server and developer's computers to guarantee that no external modification can be applied. Also, it evaluates the management of versions of the TOE, documents used during the certification and guides for the TOE, the delivery procedure and verification mechanisms for the user to verify that he/she received the certified product, so everything that the user receives is what is intended and what has been evaluated during the evaluation process.

Finally, the flaw remediation mechanisms and patch distribution can also be evaluated in some evaluations.

2.2.3.3 Development (ADV)

The objective of this evaluation task is to verify the existence and the validity of the functional specification of the TOE and its interfaces with respect to the security requirements defined in the ST.

Also, it validates if the design of the TSF implements correctly the SFR it is linked to. For that the evaluator has to validate the design of the TOE by verifying how the TOE is composed in terms of sub-systems and for higher level of assurance how these sub-systems are in their turn decomposed in lower sub-systems or modules and how each of these components interacts with each other and participates in TSFs.

Finally, a study of the security architecture of the TSF is done to analyse that it can structurally achieve the TSF desired properties and by verifying that there is no conceptual architecture flaw.

2.2.3.4 Guidance documents (AGD)

All the product guidance for final users and administrators are reviewed to guarantee that the end user can use correctly the product and more specifically can configure and use it as it has been during the evaluation.

2.2.3.5 Tests (ATE)

The developer shall demonstrate that he has sufficiently tested his product and on the other hand the evaluator shall verify the proofs provided and repeat some of the developer tests and add independent testing when deemed appropriate (i.e., too few developer tests, not tested parameters or function, etc.).

2.2.3.6 Vulnerability assessment (AVA)

In the context of CC evaluation, the goal of this task is to identify potential vulnerabilities using all information gained during the evaluation. The exploitation of those potential vulnerabilities is tested for an attacker with different resources and competences as defined by the CC for each AVA_VAN level.

2.2.3.7 International recognition

The strong benefit of the CC evaluation is its international recognition. In fact, it is the only cyber security evaluation framework whose certificates are officially recognized by the 27 signatories of the Arrangement on the Recognition of Common Criteria Certificates, in the field of Information Technology Security [21].

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- 17 certificates emitting members:
  - Australia, Canada, France, Germany, India, Italy, Japan, Malaysia, Netherlands, New Zealand, Norway, Republic of Korea, Spain, Sweden, Turkey, United Kingdom, United States
- And 10 certificates consuming members:
  - Austria, Czech Republic, Denmark, Finland, Greece, Hungary, Israel, Pakistan, Qatar, Singapore.

Since 2012 and the last edition of the CCRA, the recognition is limited only to evaluation certificates up to the assurance level 2. This is in fact a great limitation. Before 2012, the CCRA included certificates up to EAL 4. But recently North America leaded the decrease of the certificate recognition scope for political and strategic reasons. Still, it is the only widely recognized framework even for low assurance level.

Furthermore, higher evaluation levels are still internationally recognized by the European countries under the SOGIS agreement:

- Austria, Finland, France, Germany, Italy, the Netherlands, Norway, Spain, Sweden, and United Kingdom;

Under this agreement, the officially maximum certification level (mutually recognised) goes up to EAL4.

2.2.3.8 Certified Products

The list of all publicly certified products is available at https://www.commoncriteriaportal.org/products/. Those products are divided in 14 categories covering most of IT security products (as of March 2016):

- Access Control Devices and Systems – 68 Certified Products
- Biometric Systems and Devices – 3 Certified Products
- Boundary Protection Devices and Systems – 90 Certified Products
- Data Protection – 64 Certified Products
- Databases – 27 Certified Products
- Detection Devices and Systems – 16 Certified Products
- Smart Cards and Smart Card-Related Devices and Systems – 895 Certified Products
- Key Management Systems – 26 Certified Products
- Multi-Function Devices – 134 Certified Products
- Network and Network-Related Devices and Systems – 203 Certified Products
- Operating Systems – 95 Certified Products
- Other Devices and Systems – 276 Certified Products
- Products for Digital Signatures – 87 Certified Products
- Trusted Computing – 7 Certified Products

The numbers of certified products are the ones referenced in December 2015 as indicated on the common criteria portal web site. These products are either hardware or software. The largest number of products is from the domain of Smart Cards but many types of software are also certified (e.g., OS, firewall, signature products, databases, etc.).

2.2.4 FIPS

One of the other few wildly known certification scheme is the FIPS scheme. The Federal Information Processing Standards (FIPS) are public standards developed by the United States federal government. They have been issued to establish US requirements to ensure some computer security and interoperability.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Contrary to other frameworks, such as ITSEC, CC or the French CSPN, FIPS evaluations do not need the specification of a security target. The list of functions and tests to be done is known in advance and defined within the FIPS. This certification only concerns cryptographic products.

FIPS 140-2 [1] specifies four security levels for each of the 11 requirement areas. Each security level offers an increase in security requirements over the preceding level. These four increasing levels allow cost-effective solutions that are appropriate for different degrees of data sensitivity and different application environments.

Each level is an augmentation of the preceding one.

Security Level 1

Basic security requirements are specified for a cryptographic module (e.g., at least one Approved algorithm or Approved security function shall be used). No specific physical security mechanisms are required beyond the basic requirement for production-grade components.

Security Level 2

The security level 2 adds constraints for the physical security mechanisms. A requirement for tamper-evidence is made which includes the use of tamper-evident coatings or seals or for pick-resistant locks on removable covers or doors of the module physical access to protect the access to plaintext cryptographic keys and critical security parameters (CSPs) within the module.

Also, a minimum, role-based authentication is required.

The certification at level 2 requires the evaluated software and firmware components to be executed by an operating evaluated EAL2 for requirements specified in the CC Protection Profiles (PPs) listed in the standard annex. So FIPS certified products can be used only with CC certified products, making FIPS evaluation not standalone.

Security Level 3

At this level, the required physical security mechanisms are intended to have a high probability of detecting and responding to attempts at physical access. The physical security mechanisms may include the use of strong enclosures and tamper detection/response circuitry that zeroizes all plaintext CSPs when the removable covers/doors of the cryptographic module are opened.

Security Level 3 allows the software and firmware components of a cryptographic module to be executed on a general-purpose computing system using an operating system that is evaluated at the CC evaluation assurance level EAL3 (or higher) with the additional assurance requirement of an Informal Target of Evaluation (TOE) Security Policy Model (ADV_SPM.1).

Security Level 4

Security Level 4 is the highest level of certification defined in this standard. This certification level requires that the physical security mechanisms provide a complete envelope of protection around the cryptographic module with the intent of detecting and responding to all unauthorized attempts at physical access. Penetration of the cryptographic module from any direction has a very high probability of being detected, resulting in the immediate zeroization of all plaintext CSPs.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

This approach thus provides easy to understand and repeatable tests suites, but is directly limited by the scope of the tests and the products that fit in this scope. Mainly, FIPS certification suits products providing cryptographic functions and services. Other types of product cannot benefit from this certification scheme.

Also, since the tests requirements are defined within the standard, they age with it and the standard has to be rewritten every time a new security paradigm appears (i.e., new threats, new needs, etc.). Due to this reason, the standard itself states that it will be reviewed every five years.

2.2.4.1 Tests

One aspect of the FIPS 140-2 is that it defines a normalized test suite. The functional scope of the tests verifies the following objectives for the tested product:

- To employ and correctly implement the Approved security functions for the protection of sensitive information.
- To protect a cryptographic module from unauthorized operation or use.
- To prevent the unauthorized disclosure of the contents of the cryptographic module, including plaintext cryptographic keys and CSPs.
- To prevent the unauthorized and undetected modification of the cryptographic module and cryptographic algorithms, including the unauthorized modification, substitution, insertion, and deletion of cryptographic keys and CSPs.
- To provide indications of the operational state of the cryptographic module.
- To ensure that the cryptographic module performs properly when operating in an Approved mode of operation.
- To detect errors in the operation of the cryptographic module and to prevent the compromise of sensitive data and CSPs resulting from these errors.

Each product, depending on which cryptographic functions it implements, has to verify those objectives to be certified.

To verify this, the standard specifies checks to be done, e.g.:

> […]
>
> - a block diagram depicting all of the major hardware components of a cryptographic module and component interconnections, including any microprocessors, input/output buffers, plaintext/ciphertext buffers, control buffers, key storage, working memory, and program memory
>
> […]
>
> - If the cryptographic module allows operators to perform maintenance services, then the module shall support the following authorized role:
>   - Maintenance Role: The role assumed to perform physical maintenance and/or logical maintenance services (e.g., hardware/software diagnostics). All plaintext secret and private keys and unprotected CSPs shall be zeroized when entering or exiting the maintenance role.
>
> […]

Also for each type of cryptographic mechanisms it has to ensure that the cryptographic module performs properly when operating in an approved mode of operation, the NIST provides normalized tests suites for describing the input and output parameters of the test, e.g.:

```
# CAVS 11.1
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

```
# Config info for aes_values
# AESVS MCT test data for CFB8
# State : Encrypt and Decrypt
# Key Length : 256
# Generated on Fri Apr 22 15:11:50 2011

[ENCRYPT]

COUNT = 0
KEY = 7c046546c5542ff9c06823cc78efc28e8fd1e8ffd56ffc36192c6a40402c530a
IV = ea42a2fb73b36b8951c187a10205fcc4
PLAINTEXT = b9
CIPHERTEXT = 5a

COUNT = 1
KEY = 51b5ee2909a4b98eab6ef1bf8d4ae4c36b0484bf1da5240ee37b52cc40533650
IV = e4d56c40c8cad838fa57388c007f655a
PLAINTEXT = 4d
CIPHERTEXT = 60
[…]
[DECRYPT]
[…]
COUNT = 98
KEY = 751ee7fa2a48904b69ddb95998339e8ce3f5300429d92a4ce4cfaa7b327f2310
IV = f85f02e088c81b186bf34590058a76a0
CIPHERTEXT = 51
PLAINTEXT = 4c
COUNT = 99
KEY = 68e8e8dd755fb651231513c79405e71419c31d1b6d69eac2ce31cef53cd9705c
IV = fa362d1f44b0c08e2afe648e0ea6534c
CIPHERTEXT = 98
PLAINTEXT = a4
```

These lists of functional tests named Known Answered Tests (KAT) by the NIST are formalized in “rsp” files that can be found on the NIST website (http://csrc.nist.gov/groups/STM/cavp/).

Thus, the tests are more conformity tests than real security tests. As long as the functions or required justification elements exist, the test is passed and the certification granted. Thus, the evaluation of the security is directly limited by what the standard defines. If threats change or if the function can be bypassed in some ways is not directly tested.

Also, if the tests (i.e., the check list) get out-dated, the standard has to be rewritten.

The whole certification testing phase can take from 6 to 12 months, depending on the number of security modules to be certified in the product. In fact, cryptographic tests take times and usually tools have to be developed to pass all the KAT tests to be adapted to the specific product API.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 732319

2.2.4.2 Certified products

The list of certified products can be found also on the NIST website (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm). More than 2600 certificates have already been delivered covering a wide range of product providing cryptographic modules:

- HSM
- Broadband Ethernet Radio (including encryption modules)
- Network encryption modules
- Firewalls (including VPN functionalities)
- SSL modules
- SSH modules

2.3 French specific evaluation schemes

France is an interesting case study on the matter. Not only because of the French partners involved in the project but because they developed one of the very few – the only one so far to our knowledge – national evaluation scheme i.e., the CSPN. Also, France has been one of the first country to enforce ITS product security evaluation during the preliminary steps of the deployment of the afterwards abandoned Eco Taxe. Back then, some electronic tolling on board units and road side equipment had been certified.

In France, the IT security certification scheme is defined by the law [22]. This decree identifies the different actors involved in the certification process and their roles:

The ANSSI has defined that two types of products evaluations are recognized in France:

- Certification de Premier Niveau de Sécurité (CSPN)
- Common Criteria certifications.

The CSPN is a French only certification scheme. It is not recognized abroad. The complete procedure has been defined by the ANSSI itself, cf. section 2.3.1. And the Common Criteria certification scheme is based on the ISO 15408 standard described in section 2.2.3.

Furthermore, two additional kinds of product and system evaluations are defined by the “Référentiel Générale de Sécurité” (RGS) written by the ANSSI:

- Product qualification
- System homologation

In fact, this document further defines the application of [22] regarding the specific case of IT systems and products connecting the administration and the citizens as well as the administrations together.

The homologation is a process for a whole system that might have to include qualified products.

Certification and qualification are two schemes of product evaluation.

The first one only requires specifying a product to be evaluated and its associated security target (ST). The ST content has only to be defined by the evaluation sponsor and it can have any content regarding security functions to be evaluated. The only verification about the proposed security target in that case is that mandatory information is present (i.e., product name, version, description, security objectives and functions to be evaluated, no matter what they are, etc.) and the evaluation feasible.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 732319

In the case of a qualification, the security target must meet governmental needs as required by the RGS [23] and thus it defines that the ST has to be discussed and validated by the "Bureau de qualification" of the ANSSI.

So, for product qualification, first the evaluation sponsor sends the ST to qualification office, to discuss and validate its content and then enters the normal certification scheme chosen, CSPN or CC, discussed in the following sections.

2.3.1 CSPN

The first scheme we present is the French "Certification de Sécurité de Premier Niveau" (CSPN, first level security certification) [3]. This scheme only exists in France. This certification process wants to address a specific need not covered by other schemes. It aims at providing a proof that a product resists vulnerability testing done by accredited experts in limited time, so it provides evidence that the product resists enhanced-basic attack potential (attacker with good competences but restricted time and resources as defined by the CC).

The idea is simply to provide a certificate which states that the product has been tested during 25 days by security experts and no vulnerability has been found. The tests are conducted regarding the ST specifications and so aims at declaring that it is conform to this ST security specification (or not).

2.3.1.1 Evaluation Activities

There are four kinds of evaluation activities:

- A conformity analysis to the security target
  - Including documentation review
- A vulnerability survey
  - Survey of security related information that could be available in the public domain for the product and its components
- Penetration testing
  - Where for each security function defined in the ST the experts try to bypass it
- Cryptographic analysis if required

In specific cases e.g. if the product is open source, a minor code review is also added.

The vulnerability tests are chosen by the security experts and reviewed by the ANSSI.

The confidence in the appropriate testing in the product is provided by the fact that only accredited laboratories are allowed to conduct the tests, and also by the review of the report produced by the expert by the ANSSI. This report should contain the details of the tests conducted together with their results. If the report is not precise enough, if the tests are not sufficient or if the results are too doubtful, more tests can be required by the ANSSI.

2.3.1.2 Certified Products

The first certificates date back to 2011.

A wide range of software products have been certified (more than 45 at the end of 2016).

It includes the following types of products:

- Secure data erasing
- Secure storage and electronic vaults
- Hypervisor
- Firewalls

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- IDS
- Anti-virus
- Identification and access control
- Secure communications
- Industrial Programmable Logic Controller

No OBU has been certified but the process does not forbid it.

2.3.2 The EcoTaxe Poids lourds system

One of the first ITS service already deployed in different countries and should have been subject of a specific national deployment in France for Lorries is the Electronic tolling system (ETS). A specific French tax had to be imposed on all trucks using French roads. The collection of the tax implied the deployment of a dedicated national electronic tolling system. Regarding the need of trust and the dependency on an electronic system to calculate and collect a national tax, the importance of the system implied the need to guarantee its correct functioning as well as its resilience to security threats.

The decree [24] defined a certification framework for that tax named EcoTaxe poids lourd. Since the system was to be deployed at a national scale and collect a tax, the system was sensitive and critical. Such a system had no impact on drivers' safety, but none the less its sensitive aspect implied the deployment and certification of product interesting for secure ITS deployment experimentation. The technology used for the EcoTaxe system were the same as in day 1 ITS service. It was one of the first deployments with mandatory conformance and security requirements.

For the system to be legitimate and the state to be ensured of its acceptance by the citizens or at least avoid public plea in case of system errors, the system had to be reliable and provide strong confidence that it always works correctly.

The project stopped for political reasons just before the tax started to be collected. At that time, OBUs and RSUs where certified, the system was operational and about to be approved. A lot of experience has been gained on the matter.

2.3.2.1 Evaluation Activities

There were three types of evaluation activities:

- Overall system approval by accredited bodies
- OBU and RSU tests
    - Conformance evaluation defined in:
        - ETSI EN 300 674-1, ETSI EN 300 674-3-1, ETSI ES 200 674-1, ETSI TS 102 486-1-2, ETSI TS 102 486-2-2, ETSI TS 102 486-2-3, NF EN 15876-1, NF EN 15876-2, ISO/TS 13143-1, ISO/TS 13143-2.
    - Security evaluation by ITSEF

For the security test, the CESTI Oppida had been required to provide a test plan that was reviewed by the COFRAC. These test plans have then been conducted by the CESTI on the OBU and RSU. It consisted mainly in man in the middle attacks on the DSRC protocol.

The overall approval of the system pertained to the audit of its architecture and operational configuration.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

2.3.2.2 Certified Products

During the certification process and before the project ended, 3 OBUs and 3 RSUs have been certified over a period of more than one year.

2.4 State of the art summary

Many different approaches exist. None provides a universal consensus on how to provide security assurance. The existing approaches differ on many different aspects including:

- The definition of a security target and the type of product that can be evaluated
- Delivery of an official/recognized certificate
- The type of activities to be conducted during the evaluation
    - Functional tests, vulnerability tests, specification review, guidance review, standardized tests, etc.
- The required or officially recognized competences of the evaluator
- The environment used for the evaluation
    - Real operational environment versus laboratory environment
- Level of recognition of the evaluation result
- The cost
    - Time and monetary cost

Each of these aspects is treated differently in different approaches.

Security targets can be predefined indirectly in compliance assessment approaches or defined explicitly for a specific product with dedicated identification of the functions to be evaluated in more dedicated approaches. So, some are restrictive and impose state of the art security when others allow evaluating any type of product but with greater difficulty to impose state of the art function to be evaluated. Also, the use of dedicated security targets makes it much more difficult to compare security evaluations, since two evaluations with two different targets can't rely be compared. Even if the evaluated products are similar.

The recognition of the evaluation result can also greatly vary; it ranges from no official recognition, to national and international recognition. Regarding the chosen approach different aspects of the product life cycle are considered, providing more or less elements to get the assurance that the product meets its expected properties.

In the end, all these points directly impact the level of confidence obtained and the associated cost.

We compare in Table 2 the different approaches we have identified in this study.

We claim that the most complete approach that allows to provide the required high level of assurance and to tackle the wide range of solutions that will constitute the future ITS, is the CC. All other approaches are either too restrictive in terms of scope of product that can evaluate or they cannot provide a sufficient level of assurance. The problem of this approach is its high cost. That is why in the following sections we will try to define an evaluation framework that keeps most of the CC advantages and in the same time lower its main drawback.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Certification framework	Type of product	Certification Authority	ST	Assurance components / Evaluation scope	Evaluator	Tests on the TOE	Recognition	Assurance continuity	Duration and Cost
ITSEC	Any	National certification body	Defined by the level of evaluation	Security target evaluation, Life cycle, Development, Guidance documents, Functional Testing, Vulnerability testing	ISO 17025 accredited labs	Functional and vulnerability tests done by experts	Some EU members	Reevaluation	6 months to several years
TCSEC	With the required functions	National certification body	To be written for the product	Development, Guidance documents, Functional Testing	-		US	Reevaluation	6 months to several years
CC	Any	National certification body	To be written for the product, using CC standardized format	Security target evaluation, Life cycle, Development, Guidance documents, Functional Testing, Vulnerability testing	ISO 17025 accredited labs	Functional and vulnerability tests done by experts	CCRA signers up to EAL 2 SOG-IS members up to EAL 4	Reevaluation	6 months to several years
CSPN	Any	ANSSI	To be written for the product. Including all CSPN requirements	Guidance documents, Functional Testing, Vulnerability testing	Labs accredited by the ANSSI	Functional and vulnerability tests done by experts	France	Reevaluation	3 months
EcoTaxe	ETS OBU	French DoT	No	Functional Testing, Vulnerability testing	ISO 17025 accredited labs	Conformance tests and security tests done by experts	France	Reevaluation	1 year
FIPS	Cryptographic products	NIST and CSE	No	Development, Guidance documents, Functional Testing	Accredited as Cryptographic Module Testing laboratories by the National Voluntary Laboratory Accreditation Program.	Conformance tests	US and Canada	Reevaluation	3 months to more than one year

Table 3 Comparison of security evaluation approaches

3 The CARESEM evaluation framework

The French ISE project has proposed a CC scheme enhancement for the specific case of security evaluation of the cooperative ITS and autonomous vehicles named CARSEM (Cooperative Autonomous Road-vehicle Security Evaluation Methodology) [25]. This approach has already tried to tailor the CC evaluation scheme to the very specific automotive world. In SAFERtec, we will rely on this first enhancement of the CC for the ITS as a basis in order to go further in the development of a dedicated methodology and relevant tools. Those are expected to help us assess even more efficiently a high level of security assurance for ITS products. We also aim to introduce tools for security assurance *at system level* advancing the current state-of-the-art.

We recall here the targeted assurance level CARSEM focused on, since different level of sensitivity and threat can imply different level of confidence in a product, and thus impact the evaluation requirements. Then we present the global CARSEM approach.

The CARSEM proposal was innovative in the sense that the defined evaluation framework kept most of the CC advantages and tried to lower the involved costs and duration (i.e., the main drawbacks).

In the SAFERtec project we seek for the same goals and therefore, we render appropriate to re-use the main achievements of the CARSEM approach and further improve it generating the the SAF framework (see Section 5).

CARSEM proposes three main enhancements to the regular CC process:

- enforcement of recognized PPs and standards developed by regulation and domain consortia,
- parallelization of tasks including cite certification,
- roles redistribution and limited involvement of accredited bodies.

3.1 The targeted assurance levels and evaluation activities

In order to provide a comprehensive and easy to adopt framework, CARSEM takes the CC [26] as a basis. This standard is the only internationally recognized certification framework. The certification recognition is formalized through the Arrangement on the Recognition of Common Criteria Certificates signed by 27 countries for evaluations up to EAL2. In Europe, the SOG-IS has established a European recognition agreement up to EAL 4 evaluation for software and even higher for hardware. This serves as further evidence for the appropriatence of the SAF approach.

Cooperative ITS services are not yet at the level where their functionality suffices to take over the control of the system. The C-ITS Day1 services will only provide services to help the drivers in his/her decisions. In this study the author considered Human Driver (level 0 to 2 [27]) as a first step for which only a low assurance level is required: for these use cases, the target they considered was to provide a framework that would be nearly equivalent to an EAL 2 evaluation. This would allow the different actors to gain expertise in terms of assurance assessment, introducing good practices in development and test phases. This level of assurance is cheaper to obtain and is adapted to the envisioned threats for the safety functions (resistance to basic attacks).

For future automated driving systems (SAE level 3 to 5) the risks will be greater and a higher level of assurance should be required. In this second step, CARSEM aimed at an assurance level equivalent of EAL3+, i.e. an EAL3 evaluation plus the flaw remediation component (ALC_FLR.1) and the vulnerability assessment (AVA-VAN.3). This level corresponds to a resistance to enhanced basic attackers, meaning that high confidence is provided in the fact that the system can be hacked only by highly motivated

and skilled attackers. This profile of attacker will be treated by other means and not only technical means. The main goal is to make sure that regular attackers cannot harm the system.

Thus, CARSEM aims at providing an evaluation framework that would initially be equivalent to an EAL 2 certification confidence and then to an EAL 3+ certification confidence. SAF can more generally benefit from the other EAL provided by the CC; Since CARSEM proposes these going beyond the adapted EAL (see CARSEM), SAF is expected to identify even more tailored assurance levels.

A quick overview of security assurance family is presented in Section 5 of this deliverable. Here, we only present the assurance components and the associated levels recommended by the CARSEM proposal.

Assurance family	ASE	ALC	ADV	AGD	ATE	AVA
Day 1 assurance component	Yes	ALC_DEL.1 ALC_FLR.1	ADV_FSP.1 ADV_TDS.1	AGD_PRE.1 AGD_OPE.1	ATE_FUN.1 ATE_COV.2 ATE_DPT.1	AVA_VAN.2
Autonomous assurance component	Yes	ALC_LCD.1 ALC_DVS.1 ALC_CMC.1 ALC_DEL.1 ALC_FLR.1	ADV_FSP.3 ADV_TDS.3 ADV_ARC.1	AGD_PRE.1 AGD_OPE.1	ATE_FUN.1 ATE_COV.2 ATE_DPT.1	AVA_VAN.3

Table 4 Overview of CARSEM assurance activities

3.2 Use of recognized PP and security standards

The first pillar of the confidence provided by a CC evaluation is the ST. The choice of the SFR identified in the ST will define the global meaningfulness of the evaluation. This has to be adequate to the real security challenges that ITS systems face today and will have to face in the future. There is no formal method to write the right ST. Only experience and knowledge sharing from the domain's experts can lead to the definition of good PPs. Those PPs will then be used to enforce good evaluation ST. There is no universal way to guarantee that a PP or an ST is good, it is rather suggested to invest on the so-far gained experience.

So, the first step identified by the CARSEM process consists of editing a list of PPs that covers the complete ITS system. Then, they are to be validated and approved by the official certification authorities and regulators.

Relying on official PPs also greatly improves the opportunity to define and use security standards (API definition, tests suites, etc.) derived from their requirements. And with regulation or global evaluation framework pointing at them, they will provide a greater confidence in an ITS system to the general public. This will indicate that not only car manufacturer's efforts will guarantee the proper security but also that regulation or global consortia. CARSEM strongly recommends that the regulation and the standards they refer to are assessed by the car manufacturer and the regulators in charge of the public safety.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

However, this part is left open and no specific tool of methodology is proposed by CARSEM. In SAFERtec, the SAF proposition will go further on that topic, which is of critical importance.

3.3 Parallelization of tasks

The second enhancement is the parallelization of tasks. Due to its very well-structured evaluation framework, the common evaluation schedule is a sequential execution of tasks from the evaluation accredited laboratory (ITSEF) as depicted in Figure 1. Due to evaluation task dependencies, the ITSEF only starts a new evaluation task when the previous one is stopped. CARSEM proposition is to shorten significantly the overall process time span by implying more evaluation actors that will run in parallel the evaluation tasks. All evaluation tasks are started as soon as their inputs are available without waiting for the other evaluation tasks they depend on to be finished. Thus, even if the global evaluation efforts will increase due to the fact that inputs for an evaluation task will not only potentially impact the current evaluation task (as in the sequential approach), but also all the evaluation tasks depending on it. The global evaluation will end much earlier since it will be proportional to the longest evaluation tasks and not any more to the sum of them.

The idea proposed was to enhance the evaluation activities by parallelizing and distributing the work between the different stakeholders and actors involved in the process. Instead of having an entire process controlled by an official body and all the evaluation activities done by a single accredited body they propose to produce several “independent” tests and evaluation reports. Each report only has to be produced by appropriate trusted partners that will not all have to be accredited. The redefinition of required actors for the different tasks is further presented in the next section.

SAFERtec considers that the approach achieves a good level of confidence, providing at the same time the possibility to do more work in parallel employing increased manpower (as more partners are involved).

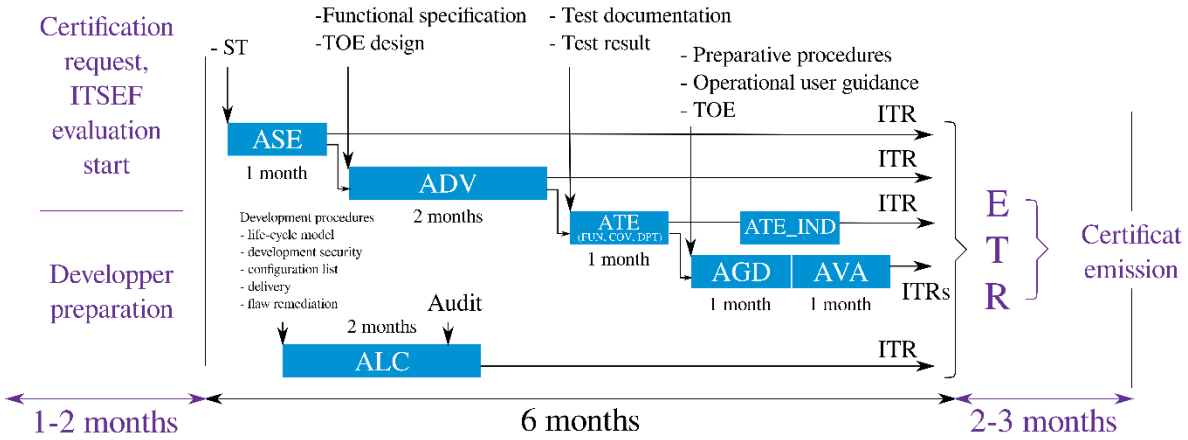

Figure 1: CC optimal evaluation schedule

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 732319

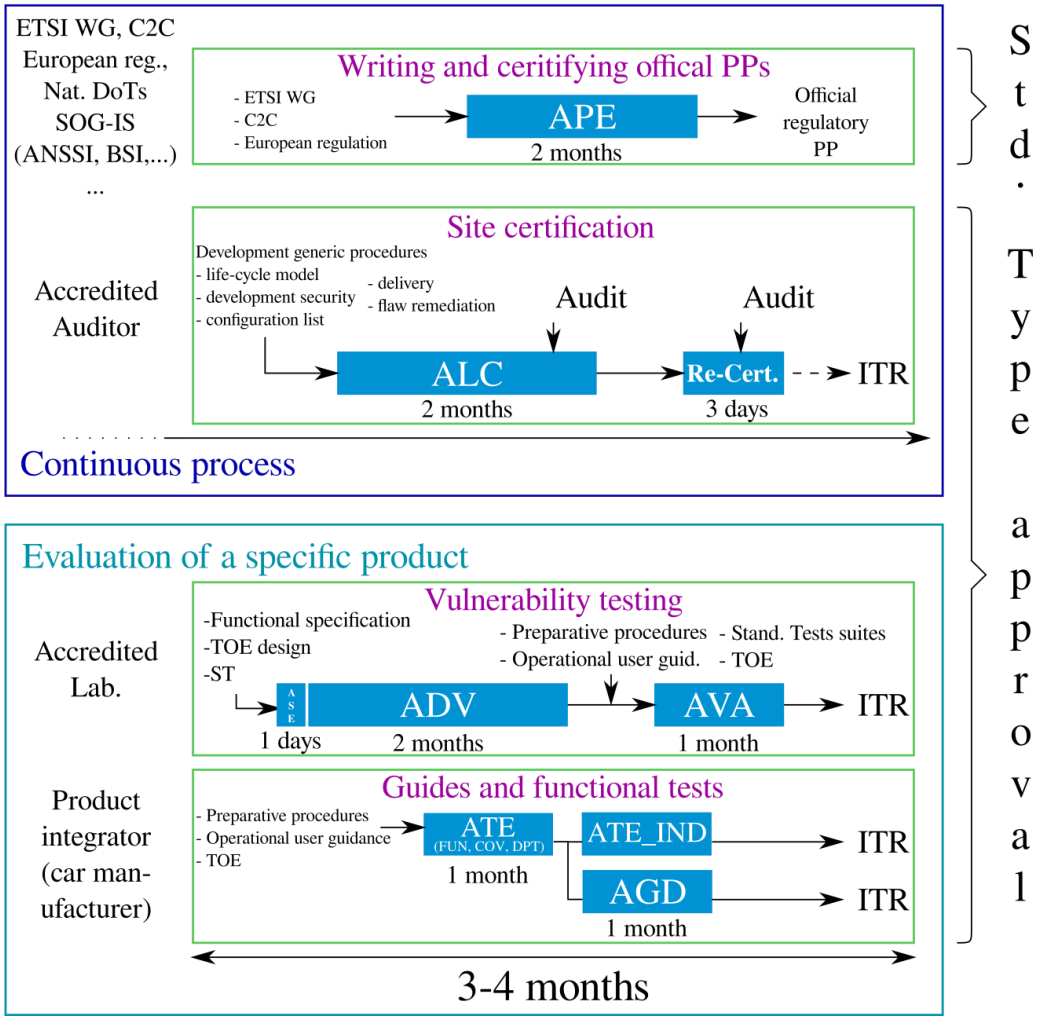

Figure 2: CARSEM evaluation process

Instead of having a single evaluation entity CARSEM proposes to rely on 4 different ones that can work in parallel to enhance the evaluation efficiency and to meet the appropriate level of assurance (see Figure 2).

Dividing the work allows to get rid of the regular sequential approach. It will also help to better integrate the ITS security evaluation in the regular car development life cycle. They argued that less effort will have to be provided for the security evaluation since already mandatory safety related integration and development best practices will be reused (evaluation tasks ALC_CMC, ALC_DEL, LC_FLR, ADV, AGD and ATE).

Furthermore, it has been proposed to generalize the site certification process already used in hardware evaluation. Thus, the evaluation of a specific product will not have to wait an ALC evaluation that is normally done for each product evaluation. On the contrary, it will be run independently for all development sites and validated for all the product they develop (within the site certification validity. In fact, best practices in the domain propose to certify a development site for a limited period (one or two years) and extend the validity with regular audits.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

3.4 Roles and actor's redistribution

One of the main aspects of our proposition is to limit the activity of external accredited bodies to very specific tasks requiring validated high level of competences. Not all the aspects of security assurance require vulnerability test competences. Some require very different skills to perform activities like life-cycle audit or developer's competences evaluation.

3.4.1 Regulators and recognized consortia

Some part of the overall confidence would greatly benefit from official bodies support. The first one is the definition of the assessment goals and the definition of the security requirements to be assessed. CARSEM greatly recommends that security evaluation should be based on the use of international protection profiles. Also, those PPs should be validated by standardization bodies and could even be referenced by international or national regulation.

3.4.2 ISO 17020 audit bodies

The automotive actors are already used to provide high safety assurance for their products. One of the main assurance vectors is through validation of safety management by car manufacturers and their tiers. In fact, it is more efficient to get confidence in a product if it can be proven that it has been designed and developed following high quality processes and if engineering competences have been audited and are confirmed to meet high standards. Those principles are not yet applied for IT security.

So far, no specific audit bodies exist for ITS, nor specific standardized audit activities. A first step is to rely on CC site certification scheme as provided by the SOG-IS. Audit activities are already identified in [28] and should follow the procedures defined in [29].

3.4.3 ISO 17025 independent security labs

Vulnerability tests and security architecture require very specific competences. It is a fulltime job and only dedicated people in specific environments can keep up with the vulnerability tests state of the art. Also, the independence of the testers and an external validation of its competences can provide assurance that the tests are correctly run. CARSEM recommends thus to have vulnerability tests done by already recognized independent laboratories that are the accredited ITSEF for CC or equivalent scheme.

3.4.4 System integrator

Finally, all the other activities do not require full independency or strong and rare competences. CARSEM aims here at optimizing the interaction with necessarily involved actors that already have the competences and have the main interest in the test results. The activities implied by ALC_CMC, ALC_DEL, LC_FLR, ADV, AGD and ATE are of interest for the integrator who will need any cases to get thoroughly through the documentation, validate the proper specification of the product for its need and also test the product for its own integration and functional validation of the entire system. Thus, CARSEM can fully rely on the system integrator or the car manufacturer if different. They will have the competence and interest in conducting the evaluation knowing that they already have to do part of those tasks anyway.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

3.5 Improve cost-benefit ratio discussion

The cost of an evaluation is twofold: on the one hand, the amount of effort needed for running the process and, on the other hand, the time to get the evaluation done.

One factor that CARSEM authors could not know for sure but estimated empirically is the increase of evaluation tasks iterations.

In fact, for each assurance component, the developer provides the required input and if problems or missing elements are found a report is produced to identify these non-conformity issues. The developer then corrects the identified elements or provides new information and the evaluator re-opens the evaluation tasks with the new inputs. The process iterates until the evaluation task is validated and the developer inputs do not change anymore.

Regular evaluation with developers already familiar with the evaluation process includes two iterations per evaluation task (i.e. one initial evaluation, plus two iterations). CARSEM assumed a new iteration average of three.

The total evaluation time is estimated higher than the sum of the developer and evaluator efforts. Developers and evaluators cannot work full time on an evaluation project. Evaluation processes are not continuous on either side. They typically work interchangeably on different projects, to avoid being idle and thus unproductive. (The developer works on the inputs, sends them to the evaluator and usually waits for the evaluator to finish its report. The developer can only work on the inputs update, after receiving the report results.)

According to empirical knowledge of CC evaluators the number of open days after which the SUCCESS report is viable, is twice the number of days of work effort for the tasks. So, if a task takes 10 days for the developer and 10 days for the evaluator (for the 3 iterations) the time after which the task will be finish is (10+10)*2=40 works days, so 2 months. .

In

		ALC_LCD	ALC_DVS	ALC_CMC	ALC_DEL	ALC_FLR	ASE	ADV_FSP	ADV_TDS	ADV_ARC	AGD_PRE	AGD_OPE	ATE_FUN	ATE_COV ATE_DPT	ATE_IND	AVA_VAN	Efforts for a product	Efforts including ALC	Cost 1	Cost 2
Product integrator (car manufacturer)	day 1						5	5			2,5	2,5	5	5	10		35	35	26250	26250
	Auton.			5	1,5	3	5	7			2,5	2,5	5	5	10		37	46,5	27750	34875
CC or SOG-IS approved ISO 17025 ITSEF	day 1								3	0						15	18	18	18000	18000
	Auton.	5	5						6	5						25	36	46	36000	46000
Developer	day 1						4	3	3		1	1	3	5	1	1	22	22	12100	12100
	Auton.	0	3	2	2	1	4	3	3	5	1	1	3	5	1	1	27	35	14850	19250
Tasks' total efforts	day 1	0	0	0	0	0	9	8	6	0	3,5	3,5	8	10	11	18	75	75	44250	44250
	Auton.	5	8	7	3,5	4	9	10	9	10	3,5	3,5	8	10	11	26	100	127,5	63750	80875
Duration	day 1	0		0			18	16	12		14		58			32			100600	100600
	Auton.	26		29			18	20	38		14		58			52			142350	181000

		ITSEF	Integrator
Total duration for products evaluation task	day 1	44	106
	Auton.	90	110

Table 5 Estimated efforts and durations in open business days (total 1 and 2 columns), CARSEM authors calculated the average time expected for each evaluation task on the line duration. Where each duration is the sum of the developer's and evaluator's efforts for this task (which results is presented in line Tasks' total efforts) multiplied by 2. As a comparison, the normal CC evaluation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

process should last about 20% less than the sum of all the tasks evaluation days. The 20% reduction corresponds to the lower number of iterations average.

A new scheme should have an evaluation time corresponding to the maximum between the evaluation time of the Car manufacturer tasks and the ITSEF evaluation tasks, since those evaluation tasks are run in parallel by the two different evaluators and the site certification also done in parallel. To calculate this, CARSEM authors added the time required for all the evaluation tasks done by each of those two evaluators (respectively red efforts cells for ITSEF and green ones for the integrator). The result is presented in the second table ("Total duration for products evaluation task). They provide a final estimation that the evaluation duration is the maximum of 44 (ITSEF evaluation) and 106 (integrator evaluation) days for Day 1, so 106 and the maximum of 90 and 110 for autonomous systems.

Thus, in both cases the outcome is about 5,5 months. A regular certification software evaluation process takes about one year pointing to an almost 50% reduction. To that should be added, the time gained not to get through certification processes (3-4 month).

		ALC_LCD	ALC_DVS	ALC_CMC	ALC_DEL	ALC_FLR	ASE	ADV_FSP	ADV_TDS	ADV_ARC	AGD_PRE	AGD_OPE	ATE_FUN	ATE_COV ATE_DPT	ATE_IND	AVA_VAN	Efforts for a product	Efforts including ALC	Cost 1	Cost 2
Product integrator (car manufacturer)	day 1						5	5			2,5	2,5	5	5	10		35	35	26250	26250
	Auton.			5	1,5	3	5	7			2,5	2,5	5	5	10		37	46,5	27750	34875
CC or SOG-IS approved ISO 17025 ITSEF	day 1								3	0						15	18	18	18000	18000
	Auton.	5	5						6	5						25	36	46	36000	46000
Developer	day 1						4	3	3		1	1	3	5	1	1	22	22	12100	12100
	Auton.	0	3	2	2	1	4	3	3	5	1	1	3	5	1	1	27	35	14850	19250
Tasks' total efforts	day 1	0	0	0	0	0	9	8	6	0	3,5	3,5	8	10	11	18	75	75	44250	44250
	Auton.	5	8	7	3,5	4	9	10	9	10	3,5	3,5	8	10	11	26	100	127,5	63750	80875
Duration	day 1	0		0			18	16	12		14		58			32			100600	100600
	Auton.	26		29			18	20	38		14		58			52			142350	181000

		ITSEF	Integrator
Total duration for products evaluation task	day 1	44	106
	Auton.	90	110

Table 5 Estimated efforts and durations in open business days (total 1 and 2 columns)

In Section 5 we will elaborate on the way that SAFERtec plans to more effectively resolve the cost-benefit trade-off.

4 Security assurance tools state of the art

Security assurance covers several aspects. The main ones are:

1. Security target definition and evaluation
2. Product life cycle evaluation
3. Product specification and architecture description
4. Functional tests
5. Vulnerability tests

In SAFERtec the tool we will propose and use mainly covers the points of product security analysis (1 and 3) and tests (4 and 5). In this section we present a quick study of the state of the art for the tools used in these two types of activities.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

4.1 Product security analysis and objectives

One of the key aspects of the assurance framework is the identification of the Evaluation Items that will constitute the basic components of the system to be evaluated. The identification of these items along with the respective functional and non-functional elements will constitute the basic parts of the assurance framework.

In the SAFERtec WP2 we have examined works from the risk analysis and security and privacy requirements engineering domains in order to map the respective research areas and identify which method best suits the project's needs. EBIOS was selected as the most adequate risk analysis method. For the elicitation and modelling of security and privacy related requirements Secure Tropos and PriS methods were selected respectively. All three methods were joined to form a unified framework for security and privacy requirements elicitation and modelling based on identified risks.

The threat and vulnerability analysis along with the elicitation and modelling of the respective security and privacy requirements that will mitigate the identified risks is a mandatory input for the assurance framework. Based on this information the Target of Evaluations (TOEs) along with the respective functional and non-functional elements will be defined.

4.1.1 Selection of the Risk Analysis Method

EBIOS® has been created in 1993 [30] and is maintained by ANSSI the French national cybersecurity agency. It was primarily intended for governmental and commercial organizations working with the Defense Ministry. Over time, it has been refined and adapted to other circumstances like industrial systems. Nowadays, it is widely used in the public sector (all the ministries and their dependent organizations), in the private sector (consulting firm, small and large businesses), in France and abroad (European Union, Quebec, Belgium, Tunisia, Luxembourg, etc.) by a lot of organisms as users or clients. It benefits from a significant experience of more than 20 years.

Document	Country	Description
ISO/IEC 27001	International	Standard that specifies the requirements of an Information Security Management System (ISMS) in terms of Confidentiality, Integrity and Availability (CIA).
ISO/IEC 27002	International	Set of good practice about ISS
ISO/IEC 27005	International	Standard that describes the risk (ISS) management process that standard and methodologies should follow
ISO/IEC 31000	International	Standard that describes the risk (general) management process that standards and methodologies should follow
Référentiel Général de Sécurité (RGS)	France	This referential aims to improve user's confidence in the online services provided by administrative authorities, especially when they deal with personal data.

List of standards and documents with which EBIOS is compliant

This methodology is fully compliant with the last version of ISO 27001, 27002, 27005 and 31000. It can be used during the very first stage of the conception of a system or on an existing system, considering the existing security measures. This methodology is adaptable and can be adjusted to the studied context. It is also possible to use certain of its parts (i.e., not the whole approach) making it more easily incorporated with other methodologies.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

The EBIOS club, composed by individual experts and organisms, is supporting, enriching and developing the methodology since 2003. The club organizes periodic meetings to encourage the return of experiences, the homogenization of the practice and the user needs satisfaction. The club has also an influence role in national and international debates. ANSSI and the EBIOS club have published the version 2010 of the EBIOS methodology to consider this return of experiences and the evolution of the regulation.

Here are the main reasons why this methodology has been chosen among other methodologies:

- Quickness: the duration of the risk assessment can be adapted with regards to the depth of the study and necessary feature available.
- Comprehensive approach: A risk event occurs when some vulnerability is exploited by a threat. EBIOS is a structured methodology that enables the possibility to break down all the risk components in entities, vulnerabilities, essential assets, threats agents and others components.
- Reusable methodology: EBIOS increases the consistency and makes possible continuous risk assessment. The risk items decompositions enables continuous updates of the assessment. Similar system assessment can also be used as a basis for another one.
- Flexibility: EBIOS can be adapted for several circumstances and its tools can be adjusted while respecting the general philosophy of the methodology. It can be used either to assess a global Information system or a small system or device (e.g., web site, messaging, etc.). It can be used for whole or the part of it.
- Proven methodology: EBIOS has already been used to perform risk assessment in industrial environment that are similar to the SAFERtec area of exploration.

A list of relevant online sources is presented for the sake of completence:

- https://www.ssi.gouv.fr/guide/ebios-2010-expression-des-besoins-et-identification-des-objectifs-de-securite/
- https://www.ssi.gouv.fr/uploads/2014/11/RGS_v-2-0_Corps_du_texte.pdf
- https://www.ssi.gouv.fr/uploads/2011/10/EBIOS-PlaquetteMetho-2010-04-081.pdf
- https://www.club-ebios.org
- https://www.club-ebios.org/site/presentations/ClubEBIOS-2010-03-23-GRALL.pdf
- https://www.cases.lu/la-methode-ebios.html
- https://www.club-ebios.org/site/presentations/ClubEBIOS-2016-07-04-VANCAUTER.pdf
- https://www.thalesgroup.com/en/critical-information-systems-and-cybersecurity/news/new-security-methodology-connected-cars

4.1.2 Selection of the Requirements Engineering methodologies for reasoning about Security and Privacy

4.1.2.1 Security Requirements Engineering Methodologies

SQUARE (Security Quality Requirements Engineering) methodology [31] is a risk-driven method that supports the elicitation, categorisation, prioritisation and inspection of the security requirements through a number of specific steps. It also supports the performance of risk assessment to verify the

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

tolerance of a system against possible threats. The method outputs all necessary security requirements that are essential for the satisfaction of the security goals of a system. The methodology introduces the concepts of security goal, threat, and risk, but does not consider the assets and the vulnerabilities of a system. All the needed security requirements should be identified by the requirements engineering team and the relevant stakeholders.

After the realisation of the importance of privacy during the development of software systems, the authors of SQUARE methodology adapted their approach accordingly, to support the elicitation of privacy requirements at the early stages of software development process [32]. The extended framework follows the same steps as the first approach of SQUARE methodology and it also integrates a technique that supports the elicitation and prioritisation of privacy requirements, namely Privacy Requirements Elicitation Technique (PRET) [33].

In [30] the authors propose Model Oriented Security Requirements Engineering (MOSRE) framework for Web Applications which considers security requirements at the early stages of the development process. It covers all phases of requirements engineering and suggests the specification of the security requirements in addition to the specification of systems requirements. The objectives, stakeholders, and assets of the Web application are identified during the inception phase. The final security requirements are elicited after a sequence of actions that include the identification – categorisation – prioritisation of threats and system vulnerabilities, the risk assessment process, the analysis and modelling, and finally the categorisation – prioritisation – validation of the final security requirements.

Another approach is the Security Requirements Engineering Framework (SREF) [34] which enables the elicitation and analysis of security requirements. This framework includes four stages. First, it identifies functional requirements and afterwards, the security goals. Continuing, it identifies the security requirements of the functional requirements. Each security requirement satisfies one or more security goals. After these steps, the framework verifies if the system satisfies the security requirements.

The authors in [35] introduced an asset-based approach for the elicitation of security goals from business process models which are then translated into security requirements. This method follows a sequence of steps. During the first step an early analysis is performed that identifies the business assets that are valuable and must be protected against security risks. The second step is dedicated to the elicitation of security requirements during the examination of the security risk of business assets. The final stage is the elicitation of security requirements which results in the generation of business rules that satisfy security goals of the system under examination.

A well-known goal-oriented requirement engineering approach, KAOS [36] was introduced for the elaboration of requirements from high level goals. Exceptional agent behaviours, namely obstacles, were responsible for the fulfilment of goals. These obstacles were identified and resolved through the elaboration of scenarios between software and agents, responsible for the production of a reliable system [37] [38]. The KAOS methodology has been extended [39] in order to elaborate security requirements as well. The output of this extension is the development of two models. The first model corresponds to the system-to-be, aiming to describe the software and the relations between goals, agents, objects, operations, and requirements. The second model, which is regarded as an `anti-model', captures possible attackers, their goals, as well as system vulnerabilities, in order to elicit

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

potential threats and security requirements for the prevention of these threats. The aforementioned security requirements that the anti-model derives are regarded as countermeasures and are integrated to the first model.

The work in[40] [41] proposes the Problem-based Security Requirements Elicitation (PresSuRE) Methodology that facilitates the identification of security needs during the requirements analysis of software systems. More specifically, it provides a computersecurity threat recognition and then the development of security requirements. This methodology uses problem diagrams to support the modelling of functional requirements. Firstly, based on its contents, this methodology identifies system's assets and the rights of authorised entities. Then, it determines possible attackers and their abilities. Based on these steps, PresSuRE generates graphs which depict threats on system's assets. Every functional requirement of each asset is related with possible threats and security requirements.

Secure Tropos Requirements Engineering Methodology was introduced [42] in order to cover system requirements during the whole software development process. However, Tropos methodology gives a strong focus on the early stage of system analysis. The framework includes five development phases: early requirements, late requirements, architectural design, detailed design and implementation. However, security concepts have not been considered in any of these phases. Thus, Mouratidis et al. extended Tropos methodology in order to accommodate security concepts during the requirements analysis. The extension is called Secure Tropos [43] and utilizes only the early and late requirements phases of Tropos framework. Secure Tropos introduces the concept of security constraints. According to the authors, security constraints are a set of conditions, rules and restrictions that are imposed on a system and the system must operate in such way that none of them will be violated [43]. In the early requirements phase, a security diagram is constructed in order to represent the connection between security features, threats and mechanisms that help the satisfaction of security goals. The security diagram is taken into consideration at the late requirements phase in order for the designers to impose security constraints to the system-to-be. The enforcement of security constraints in different parts of the system can facilitate the disclosure of possible conflicts between requirements.

4.1.2.2 Privacy Requirements Engineering Methodologies

In the area of privacy requirements, in [44] the authors present LINDDUN, a privacy threat analysis framework which, in its first release, aimed at the elicitation and fulfilment of privacy requirements in software-based systems. The process that LINDDUN follows is that a data how diagram (DFD) of the system is designed and then the identified privacy threats are related to DFD elements. Privacy threat trees and misuse cases are used for the collection of threat scenarios that might affect the system. Moreover, this methodology supports the elicitation of the final privacy requirements and the selection of appropriate privacy enhancing technologies. The final stage of this methodology is the prioritisation and validation of privacy threat through risk assessment. LINDDUN also provides a map that connects privacy enhancing technologies with each privacy requirement, facilitating thus, the system designers to select the most appropriate techniques that are able to satisfy privacy requirements.

Next, in [45] the authors adopt the concepts of privacy-by-policy and privacy-by-architecture and propose a three-sphere model of user privacy concerns, relating it to system operations (i.e. data transfer, storage and processing). Additionally, the Modelling and Analysis of Privacy-aware Systems

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

(MAPaS) framework [46] is a framework for modelling requirements for privacy-aware systems. The ABC4Trust project [47] protects privacy in identity management systems.

Privacy Safeguard (PriS) [48], a privacy requirement engineering methodology, incorporates privacy requirements into the system design process. The methodology PriS aims to cover the gap between system design and implementation phase. PriS considers privacy requirements as organisational goals and through the use of privacy-process patterns in order to describe the impact of privacy goals to the affected organisational processes. The next step is the modelling of the privacy-related organisational processes. These processes aim to support the selection of the system architecture that best satisfies them.

Out of the above broad set of requirement engineering methodologies, SAFERtec has introduced an innovative combination of EBIOS, SecureTropos and PriS, as explained in D2.2 and D2.3

4.2 Functional and security tests tools

As for the exact same reasons that there is no universal evaluation framework for IT and ITS security, there are no universal security tools. Tools required to test security functions of ITS elements and systems are fully dependant of the specific details of their implementation (e.g., API, HMI, technologies, etc.). Thus, to test either their functionally and/or make vulnerability tests we will rely on the already existing and most widely used testing tools.

Since there is no known benchmark or detailed characterisation study of testing tools, we only provide here a list we think relevant; the list is by no means exhaustive:

- Scanning tools
  - Nmap, https://nmap.org/
  - Masscan, https://github.com/robertdavidgraham/masscan
  - Nexpose, https://www.rapid7.com/products/nexpose/
  - Nessus, https://www.tenable.com/products/nessus/nessus-professional
  - Dirbuster, https://sourceforge.net/projects/dirbuster/
  - Dirb, http://dirb.sourceforge.net/
  - Findsploit, https://github.com/1N3/Findsploit
  - Sslyze, https://github.com/iSECPartners/sslyze
- Vulnerability tests and exploitation tools
  - Web
    - Arachni, http://www.arachni-scanner.com/
    - Burp, https://portswigger.net/burp/
    - Parameth, https://github.com/mak-/parameth
  - Fuzzing and brute force
    - Wfuzz, https://github.com/xmendez/wfuzz/
    - Patator, https://github.com/lanjelot/patator
    - Hydra, https://github.com/vanhauser-thc/thc-hydra
    - Peach, https://www.peach.tech/
  -
- Network analysis, interception and packet manipulation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

  - Wireshark , https://www.wireshark.org/
  - Tcpdump, http://www.tcpdump.org/
  - Netzob, https://github.com/netzob/netzob
  - Ettercap, http://www.ettercap-project.org/ettercap/
  - Dsniff, https://www.monkey.org/~dugsong/dsniff/
  - Netsed, http://silicone.homelinux.org/projects/netsed/
  - Scapy, https://scapy.net/
  - Packet Sender, https://packetsender.com/
  - Haka, http://www.haka-security.org/download/haka.html
- Exploit
  - Metasploit, https://www.metasploit.com/
  - Canvas, https://www.immunityinc.com/products/canvas/
  - Core impact, https://www.coresecurity.com/core-impact
- Radio
  - GNU Radio, https://gnuradio.org/
- For everything else
  - Human brain and C, C++, java, python, bash, etc.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

5 The SAFERtec assurance framework

In this deliverable we present the global assurance framework defined by the SAFERtec project. Each work packages of the project contributes to the assurance assessment of the use cases defined and developed in the project. This assurance framework combines all the different efforts made in all the project's work-packages in one global assurance approach. Some participated to the framework itself, and others are only used to validate the approach. In both cases they will help to define and refine tools and methods aiming to provide higher and cheaper security assurance to ITS systems.

The SAFERtec Assurance Framework (SAF) relies on already existing approaches, standards and tools combined in an innovative way to achieve a high assurance level *at system level* for ITS systems. As presented in the previous section, many approaches and related works have been done in the past in order to evaluate security. None of them receives a global consensus on its efficiency. They are all either too expensive or provide questionable results. The only framework that provides strong and recognized assurance results is the CC. But this approach is long and expensive and for this reason *does not scale well at system level*. This is partly due to the fact that it is a generic framework that asks experts and developers to provide very specific product documentation (at version level) and thus shape verification methods (for the evaluator). This is very true for software and a bit less for hardware module. The difference is mainly based on the fact that the latter are much more standardized and much more similar with each other. Thus, many elements are similar and easily reusable or redo-able for different evaluations. This is what we will try to enforce for ITS systems. We will try to standardize and provide as many re-usable elements as possible to implement and prove ITS-adapted security requirements. In that sense, we aim at re-using the CC in a system-limited and dedicated way to gain higher and cheaper assurance than the normal approach would do.

As presented in section 2.1, the different IT security evaluation methods all have to deal with three main dimensions, either directly formalized or not:

- What has to be evaluated?
- Which evaluation activities?
- Who is competent and has to be in charge of what?

The state-of-the-art shows that the second point is well defined by the CC in a way that is generic enough to fit any product. The CARSEM approach greatly enhances the last point and brings the CC evaluation into the context of the automotive domain. Thus, two points remain to be enhanced in order to benefit from the full CC process at affordable efforts; introducing relevant and efficient STs as well as specific tools and methods to produce element of proofs for the developer and run evaluation tasks for the evaluator.

We first provide a global overview of SAF. Then, we further detail for each assurance task how SAFERtec results will enhance each of those evaluation tasks. The specific details of each improvement are to be found in the different SAFERtec work packages and deliverables pointed here.

5.1 General overview

The ISE project proposed a first refinement of the common CC certification approach for ITS product evaluations. This framework elies on the regular CC evaluation tasks but proposes to execute them in

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

parallel. This approach is based on the fact that different actors can be identified to work in parallel on the different evaluation tasks; the idea which is an important enhancement in terms of time and cost efficiency to run the complete CC evaluation process.

SAF proposes to carefully enhance this concept by providing dedicated tools and knowledge bases to *ease* and *speed up* either the developer's input production or the relevant evaluation tasks. In fact, for every evaluation tasks the CC defines specific inputs to be provided by the developer together with the corresponding specific evaluation points to validate those inputs. The ISE framework does not address this concrete part of the CC evaluation process. It only redefines an evaluation schedule and relevant responsibilities to adapt them to the ITS world. As presented in Figure 3, SAFERtec on the one hand tackles the enhancement of CC input production and evaluator tools and on the other issues specific ITS oriented guides to accelerate these tasks.

Thus, SAF serves as a significant enhancement of previous approaches (i.e., the CARSEM evaluation approach) providing specific tools as presented in Figure 3. In this figure, for each CC evaluation task, we present the prosed enhancement in the "developer" and "evaluator" boxes (e.g. for the CC ADV evaluation task SAF provides the WP2 model methodology to help developers to provide more appropriate system architecture descriptions to be used as input for this evaluation task). Also, we add to regular CC framework an extra assurance component named AOP for OPerational Assurance component, which will provide operational assurance metrics to be run in the complete running ITS system to demonstrate security of the *global system* in its operational environment. That is something that the CC approach so-far lacks.

Thus, as presented in Figure 3 SAF aims at enhancing the security target definitions and evaluation (ASE/PP), by providing security-requirements libraries to be compiled in protection profile in the WP2 work. This will greatly help developers to write faster relevant and widely recognised security targets for their products. Also, the system model tools together with the reference architecture and interfaces knowledge base (to be realized in WP6) will greatly help developers to provide precise and well-structured architecture description and demonstration of their product security resilience; this will serve as input for the specification and architecture description evaluation task (ADV). Finally, WP5 and 6 will provide specifications and tools that will help run tests on ITS products, which will help both the developer and evaluator in evaluation tasks that relate to functional and security tests (ATE and AVA).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

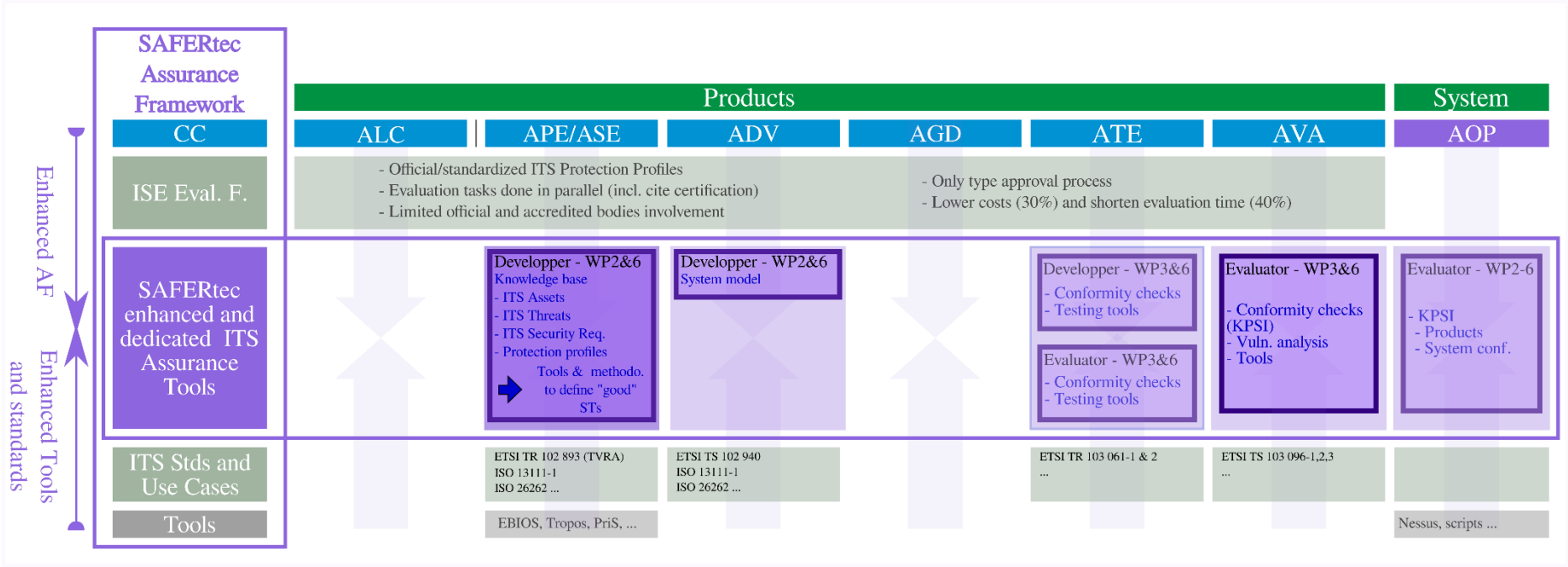

Figure 3 The SAF overview

5.2 ALC (evaluation task)

5.2.1 Evaluation task objective

This assurance component has one main goal:

- To guarantee to the user the integrity of its product:
  - the code of the product has not been tampered
    - limiting the possibility of installation of back-doors by attackers without the knowledge of the developer (in the TOE source code)
  - the code tested during the evaluation is the same as the one received by the end user

To assess this, this assurance family is composed of 5 elements evaluating:

- ALC_LCD: the life-cycle management of a product, i.e., development plan describing the way that specification, conception, coding, tests, etc. are handled by the developer.
- ALC_DVS: the physical and IT security of the development server and developer's computers
- ALC_CMC: the management of versions of the TOE, documents used during the certification and guides for the TOE
- ALC_DEL: the delivery procedure and verification mechanisms for the user to verify that he/she received the certified product
- ALC_FLR: the flaw remediation mechanisms and patch distribution

5.2.2 SAFERtec enhancement

The CARSEM approach has already validated the interest of having ALC evaluation; we render any further enhancement as not needed. Securing and correctly managing a TOE development is not something that can be standardized. Developers have the right to deal with these challenges the way they prefer. The evaluation of these elements is already (sufficiently) formalized by the CC and existing documents on the matter. No specific tools or reference elements can be defined for ITS developers thus this evaluation task enhancement is out of the SAF scope.

5.3 ASE/APE (evaluation task)

5.3.1 Evaluation task objective

One of the main elements of any relevant security assurance evaluation is the *Security Target* (ST). This document must contain the set of *Security Functional Requirement* (SFR) that has to be validated. It defines for the all evaluations what has to be evaluated. This is the objective of the project.

This evaluation task has to be done before any other since it is the starting point and specification of the evaluation objectives.

This document will define all the evaluation objectives and the chosen test environment as well as the relevant hypothesis. Thus, this document will define all the relevance of the final result and the corresponding assurance gained.

A ST contains information about the TOE and specifications of evaluation's parameters, among which:

- An introduction
  - *Target Of Evaluation (TOE) reference (exact evaluated version)*

    - The TOE overview (general description of the product type and its main security features)
    - The TOE description
        - The precise description of the TOE environment to be used for the evaluation
        - A more detailed description of the product and its main functions (several pages)
- The definition of the security problem considered (used to justify the choice of security function to be evaluated)
    - Considered threats
    - Security policies justifying security functions
    - Assumptions on the TOE and its environment
- Security objectives for the TOE and its environment and the rational justifying how it matches the security problem
- Security requirements
    - The list of SFRs to be evaluated
- TOE summary specification
    - The description of the TOE functions and how they satisfy all the SFRs

CC also requires more specific information such as:

- Conformance claims
    - The list of declared conformance used for the evaluation (to the different CC parts and versions as well as to certified PPs, when it is the case)
- The list of assurance components to be applied during the evaluation (specific to the CC).

This assurance task has two main objectives: define and validate the correct and precise formalization of the evaluation objectives. It is very important to understand what these objectives for a certification are in order to understand the correct meaning of the evaluation result.

An evaluation applies only to the specific version of a product, identified clearly and unambiguously in the ST. It validates only the security functions identified by the SFR and only under the conditions defined in the ST, i.e.:

- for the specific test environment
- under the hypothesis of the security problem

Validating the suitability or relevance of the ST is not included in its objectives. This has to be done by other means in the evaluation framework, such as validation by a community or designated trustworthy security experts.

Thus, it is of tremendous importance to have a correctly defined ST. This is what ASE is about: verifying that the ST contains all the fundamental elements to define the evaluation objectives and

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

assumptions. However, this task does not and cannot aim at validating the meaningfulness or the relevance of the chosen security requirements.

There is no way to enforce meaningfulness in an ST, other than having it acknowledged by as many security experts as possible. This is why the CC has defined protection profiles and the corresponding way to evaluate them (APE). Protection profiles are generic STs that apply to a whole range of products (e.g. firewalls, VPNs, routers, etc.). They more or less contain the same elements but are written in a more generic way (there is no specific product version identified, test environment or implementation details). This way, they allow gathering experts' knowledge on relevant security objectives and requirements for the specific product class. PPs are meant to be written and acknowledged by expert's communities or even sometimes regulation. They help to gather empirical knowledge on what relevant security concerns have to be addressed and how. Thus, they greatly help to enhance evaluation with ST conformant to widely approved PP. Thus, APE is used to validate PP to be used as reference for ST writings of important product classes.

5.3.2 SAFERtec enhancement: developer inputs

SAFERtec enhancement regarding PPs and STs writing and evaluation is twofold:

- To provide a methodology that helps to define better ITS PPs and STs in terms of relevance and precision by defining more precise threats and security countermeasures for them (SAFERtec WP2 – Deliverable 2.2 and 2.3)
- To apply this methodology in order to provide reference security requirements and eventually PPs for some major ITS elements (WP2 – Deliverable 2.4 & WP6 – Deliverable 6.2)

5.3.3 SAFERtec enhancement: evaluator activities

SAFERtec plans to introduce a modular PP for the central part of the ITS environment i.e., the connected vehicle; this is expected to greatly ease the ST evaluation of ST conformant to those PPs. STs evaluation is typically faster when they are fully conformant to good and properly written PPs.

5.4 ADV (evaluation task)

ADV is further decomposed into 3 assurance families: FSP, TDS and ARC. We describe their different objectives individually since we propose to enhance all of them.

5.4.1 Evaluation task objective

5.4.1.1 Functional specification (ADV_FSP)

The objective of this evaluation task is to verify the existence and the validity of the functional specification of the TOE and its interfaces with respect to the security requirements defined in the ST.

This evaluation task verifies the proper identification and specification of the different external interfaces:

- HMI,
- API,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- network interfaces,
- physical interfaces,

This identification only concerns the implementation of the different TOE functions and more specifically, the security functions of the product.

This is useful to guarantee that the developer can demonstrate that the product really satisfies the ST and that the coverage of the SFR defined in the ST is correct; later on, it helps to make sure that the functions specified in this architecture are in fact implemented in accordance with this specification.

5.4.1.2 Security Architecture (ADV_ARC)

A study of the security architecture of the TSF is done to analyse that it can structurally achieve the TSF desired properties and to verify that there is no conceptual architecture flaw.

Without a sound architecture, the entire TOE functionality would have to be examined. Thus, this task allows verifying that the developer is able to justify that his product's TSF fulfils the SFRs in the ST.

5.4.2 SAFERtec enhancement: developer inputs

It is very difficult to have a precise and exhaustive description of all the important (expected) input and output behaviour of all products' interfaces. The tools and methodology provided by WP2 are meant to be used by the developer to fully describe and justify his product's security countermeasures. Thus, those tools, together with the methodology, force the developer to carefully and thoroughly define its product's interfaces and assets.

All the steps of the methodologies and all the mandatory data (and data structures) to be filled in those tools force a proper definition of the product. When no specific methodology is used, these tasks become much more cumbersome and highly prone to errors, inconsistencies or big lack of precision. Tools and methodologies can only enforce higher levels of quality for this document. Regarding security this is a very important parameter. Often evaluators have to accept approximated or inconsistent descriptions (i.e., they are artificially complex and at the same time not precise enough). This is because it is really hard (for the developer) to simply describe complex systems when not following any well-structured methodology, which is what typically happens. Most of the time, the tools and methodologies they use leave a great place for interpretation between high level specification and real implementation, where exactly lies the expected level of details required for the ADV evaluation task.

5.4.3 SAFERtec enhancement: evaluator activities

SAFERtec does not plan to invest efforts on improving the above evaluator activities (on this task).

5.5 AGD (evaluation task)

5.5.1 Evaluation task objective

5.5.1.1 Preparative procedures (AGD_PRE)

Here the objective is to validate that the documentation or supports provided to the TOE user allows him to transform the delivered object (cf. delivery procedures) into an operational TOE, as identified in the ST. Typically, if a security product is not correctly installed and configured it does not provide

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

security features. It can even be an additional vulnerability point for the system. Or it can dangerously provide a false impression of security whereas the security features are disabled. Many default configurations of security products result in the 'switch-off' of security functions. They have to be correctly activated and configured for the system to be operational, since default security behaviours are not compatible with most operational systems. Also, here, another goal is to verify that TOE guides allow the TOE installation and configuration by the user to verify every recommendation or constraint expressed in the ST.

5.5.1.2 Operational user guidance (AGD_OPE)

Here the objective is to be able to operate the TOE in use cases stated in the security target. Operational user guidance refers to written material that is intended to be used by all types of users of the TOE in its evaluated configuration: end-users, persons responsible for maintaining and administering the TOE in a correct manner for maximum security, and by others (e.g. programmers) using the TOE's external interfaces. The objective is to minimize the risk of human or other errors in operation that may deactivate, disable, or fail to activate security functionality, resulting in an undetected insecure state.

5.5.2 SAFERtec enhancement

SAFERtec, as above, does not plan to invest efforts on improving the above evaluator activities (on this task)

5.6 ATE (evaluation task)

5.6.1 Evaluation task objective

In CC evaluations, this class aims at verifying that the TOE and its TSF behave as described in the ST, functional specifications and TOE design. It does not contain any penetration testing, only functional testing. All the vulnerability tests are done in the AVA_VAN evaluation. Here, the goal is to validate that all the security functions work properly.

On the one hand the developer shall demonstrate that he has sufficiently tested his product, on the other hand the evaluator shall verify the proofs provided and repeat some of the developer tests or even add independent testing when deemed appropriate (i.e., very limited developer tests, not tested parameters or function, etc.).

Usually the validation of the completeness of the developer test plan is done in the same time by the evaluator since the two evaluation families ATE_COV and ATE_DPT covering this aspect require the same inputs. That is what we do in this section by considering only three evaluations activities: test execution by the developer (ATE_FUN), coverage (ATE_COV and ATE_DPT) and independent testing by the evaluator.

So, for ATE_FUN, the objective is to verify that the TSF interfaces identified in ADV_FSP have been tested by the developer, the appropriateness of the considered environment/conditions and finally, the correct documentation of this activity.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

For ATE_COV and DPT the objective is to prove that all TSFI and subsystems are covered by tests, by verifying that all TSFI have been tested thoroughly enough, i.e. testing the correctness of TOE's internal functions and interactions.

5.6.2 SAFERtec enhancement: developer inputs

ITS systems are very complex and will include a large number of different elements coming from different vendors. Two main challenges have to be addressed by the developer: To make sure that their product is fully functional (i.e., security and functional features); it operates in a way that avoids safety issues and enhances interoperability. However, to introduce a relevant complete and thorough test plan, is quite complicated. It's often a real market advantage to be able to define a test plan and develop a test bed that allows the testing of all the expected behaviour of a product. They can potentially give the confidence that no important use case or possible input has not been tested and thus all product behaviours are proven to be good.

A second very important point will be to guarantee proper interoperability. In such critical systems as future ITS systems, where functional errors can lead to safety issues, it will also be very important to provide the guarantee that all elements in the system really interoperate as expected and no critical interaction has been left untested.

Achieving this goal is very complex, especially if one relies solely on the developer's skills. It is important to avail proper tools and pieces of information to guarantee that the product has been correctly tested (in terms of depth and coverage using the proper tools).

SAFERtec plans to considerably assist the developers in those tasks by providing:

- In the deliverable D3.2 assurance metrics to quantify the trustworthiness attributes of the Connected Vehicle System will be presented. The SAFERtec metrics taxonomy will provide to the community and thus to the developer, metrics which are quantifiable, repeatable and comparable. Those metrics will help the developer to:
    - Estimate the validity of its product regarding security, privacy, reliability and safety regarding the main ITS products requirements, as identified in the SAFERtec framework
    - Validate the conformance with main regulations, policies and standards
- In the deliverable D6.2 and D6.3 a Knowledge Base and an Inference Engine for the main ITS component interfaces and functions of assurance evaluation will be introduced. Those outcomes will be based on elements of the D3.2 and the additional tests run by the developer.

Even if the metrics and tools will be more focused on security functions than ITS functions, they will provide good means to evaluate the product quality and furthermore, assess whether it fulfils its specifications. Those tools and the knowledge base will also greatly benefit from the study of the use cases and thus WP4 results will serve as a basis for their further improvement.

5.6.3 SAFERtec enhancement: evaluator activities

The same tools proposed to the developer can be used by the evaluator to assess the products assurance. Since in the assurance task the evaluator has to verify the scope and the depth of the

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

developer test plan as well as to replay some of the tests that the developer (claims he) carried-out, those exact same tools can be used by the evaluator to accelerate his work.

5.7 AVA (evaluation task)

5.7.1 Evaluation task objective

In the context of CC evaluation, the goal of this task is to identify potential vulnerabilities using all the information gained during the evaluation and to test the exploitation of these potential vulnerabilities for an attacker with different resources (depending on the AVA_VAN level).

5.7.2 SAFERtec enhancement: developer inputs

The enhancement proposed by SAFERtec for the AVA evaluation tasks is actually the same as for ATE, except that WP3 and WP6 work are even more focused on security functions testing than on functional testing. Thus, we aim to greatly help developers to test (on their own) their product and consequently to ease the evaluation process; this is to be done by validating before the evaluation that the product is in fact ready to be so.

Often in evaluation, the product is sent to the evaluator without performing all the basic security checks that are not so demanding in terms of competences but can be very burdensome or very hard to implement for an inexperienced. Such tools will enforce higher security standards for products going through an evaluation process, making the process much faster and cheaper. This is one of the main goals of SAF. Thus, SAFERtec will help developers with the same tools as specified in the previous section.

5.7.3 SAFERtec enhancement: evaluator activities

SAFERtec will provide all the aforementioned tools to assess product security assurance. This is actually the main point of WP3 and WP6. All the tests and tools provided by WP3 and WP6 will take the form of either conformity checks or security function resilience which is a form of vulnerability. Tools to be developed for those tests would be adapted to different ITS component interfaces and thus could be re-used by security experts to run more efficiently vulnerability tests.

5.8 AOP extended component

5.8.1 Evaluation task objective and definition

The objective of this task is to bring security assurance at the operational system level. In fact, CC approach only defines ways to test products in a laboratory environment. The problem is that actual ITS systems will be the composition of several independent products (developed by different companies) that will be combined into one global ITS system. Providing proves that individual components are secure in laboratories does not prove that the final global running system also is.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Many configuration problems can arise when integrating the components together and many real interactions can be oversight pointing to security breaches in the final system.

Here, we define an AOP class for the SAF, which consists of running KPIs or Key Security Performance Indicators. In this assurance class we define 3 levels. The first one consists of running partial configuration and conformity tests for secure elements and their security requirements. The second level is a complete coverage of secure elements requirements by either configuration validation or conformity checks on all its SFRs. Finally, the third requires operational evaluation of each secure element by vulnerability tests or exhaustive coverage of all SFR on all secure components using automated configuration, conformity and security tests (i.e. SKPI).

AOP_VAN.1 Vulnerability analysis

Dependencies:	ADV_ARC.1 Security architecture description ADV_FSP.2 Security-enforcing functional specification ADV_TDS.1 Basic design AGD_OPE.1 Operational user guidance AGD_PRE.1 Preparative procedures

Objectives

A partial configuration and conformity test for each secure element and their security requirements is run to verify that secure elements are correctly used in the operational system.

Developer action elements:

AOP_VAN.1.1D	The developer shall provide the ITS system for testing.

Content and presentation elements:

AOP_VAN.1.1C	The ITS system shall be suitable for testing.

Evaluator action elements:

AOP_VAN.1.1E	The evaluator shall perform for each system secure element (e.g. system element for which SFR have been defined) at least one of the following actions: • Configuration checks to very that the element is used under the state of the art security configuration recommendation. • Partial conformity test of the component TSFI

AOP_VAN.2 Vulnerability analysis

Dependencies:	ADV_ARC.1 Security architecture description **ADV_FSP.3 Functional specification with complete summary** ADV_TDS.1 Basic design AGD_OPE.1 Operational user guidance

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

AGD_PRE.1 Preparative procedures

Objectives

An exhaustive configuration and conformity test for each secure element and their TSFI is run to verify that secure elements are correctly used in the operational system. The objective of this component is to confirm that all secure elements are correctly configured and their TSFIs are operationally been tested.

Developer action elements:

AOP_VAN.2.1D	The developer shall provide the ITS system for testing.

Content and presentation elements:

AOP_VAN.2.1C	The ITS system shall be suitable for testing.

Evaluator action elements:

AOP_VAN.2.1E	The evaluator shall perform for each system secure element (e.g. system element for which SFR have been defined**) the following action:** • A configuration check to very that the element is used under state of the art security configuration recommendation. • **And a partial conformity test of independent functional test of each of the TSFI component.**

AOP_VAN.3 Vulnerability analysis

Dependencies:	ADV_ARC.1 Security architecture description ADV_FSP.3 Functional specification with complete summary ADV_TDS.1 Basic design AGD_OPE.1 Operational user guidance AGD_PRE.1 Preparative procedures

Objectives

An exhaustive configuration and security validation for each secure element and their TSFI is run to verify that secure elements are correctly used and no potential vulnerability can be exploited in the operational system. The objective of this component is to confirm that all secure elements are correctly configured and secure.

Developer action elements:

AOP_VAN.3.1D	The developer shall provide the ITS system for testing.

Content and presentation elements:

AOP_VAN.3.1C	The ITS system shall be suitable for testing.

Evaluator action elements:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

AOP_VAN.3.1E	The evaluator shall perform for each system secure element (e.g. system element for which SFR have been defined) the following action: • A configuration check to verify that the element is used under state of the art security configuration recommendation. • **And a complete conformity test at the state of the art or penetration testing to determine that the element is resistant to attacks performed by an attacker possessing moderate attack potential**

5.8.2 SAFERtec enhancement: developer inputs

No specific SAFERtec element will be developed to help the developer for this task, since all the inputs that have to be provided are already provided for the other evaluation tasks. However, the tools and methodologies developed in WP2 will help the developer to provide the definition of the secure component at the system level.

5.8.3 SAFERtec enhancement: evaluator activities

SAFERtec will provide:

- KPIs in the deliverable D3.2 for assurance metrics to quantify the trustworthiness attributes of the Connected Vehicle System that can be applied in the operational system.
- In the deliverable D6.2 and D6.3 a Knowledge Base and an Inference Engine for the main ITS component interfaces and functions assurance evaluation based on elements of the D3.2 and the additional tests run by the developer.
- From WP4 and 5 testing tools and definition of conformity checks to be applied to well defined (the more recurrent or standardized) ITS secure component

5.9 SAF Assurance level

Thanks to our approach we can reuse any of the CC assurance evaluation levels, since we fully rely on the CC. Also, thanks to our methodology proposed to write ST, we can perfectly identify the required assurance level for a specific product to be evaluated. Thus, our approach can correctly identify and handle the right assurance level assessment.

5.10 SAF Assurance continuity

The state of the art and the empirical knowledge gathered by ITSEF and certification bodies demonstrate that even the security assurance approach proposed by the CC does not provide the required level of assurance. In fact, even if the CC proposes a way to handle it, it has never been used.

Every testing laboratory knows that even the slightest modification of a product can imply a huge security impact. Only in very rare cases it is possible to demonstrate with absolute certainty that the security functions of the product have not been modified. Thus, a new version of a product cannot benefit of the evaluation of the previous one. The whole process has to be executed again. However, if the modification is minor and the developer can perfectly trace and argue about all the modifications

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

(compared to the previous version) and their impact, the global evaluation process will greatly benefit from the previous evaluation. As many information and evaluation activities can be reused, time and money related to the process can be gained.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

6 Conclusions

In this document we have presented a state of the art of existing security evaluation technics and certification frameworks.

This state of the art clearly indicates that evaluating cyber security is a costly task that requires very specific expertise. So far, it is too difficult to have fully formalized approaches or tests-suites to validate security, since products and systems can virtually take any form (i.e. provided function, architecture, technologies, etc.). That's why only generic approaches are efficient. They all require expert's work to tailor generic evaluation requirements into specific validation tasks for a very specific product or system.

However, the main reference known so far, the Common Criteria, provides a very good framework to gain the high level of assurance required by the ITS systems. This method is generic enough to be adapted to any product with any technology. Its only and main drawback is it's cost (in terms of time and money). But this can be relaxed if its scope is limited and some of the proposed evaluation tasks made more specific and/or relying on predefined knowledge.

A first enhancement of the CC framework has been proposed in the CARSEM methodology. This methodology provides a first important cost reduction by more directly relying on the strong specificities of the automotive domain, i.e. very strong competences of the car manufacturer in self-validation (used here for all the functional and guidance's documentation validation) and strong relationships with their OEM to validate in the same way their product. Lowering then the costs by allowing parallel evaluation task execution since the car manufacturer can run part of the evaluation tasks together with accredited labs that will still run sensitive evaluation tasks (i.e. tasks requiring strong confidentiality agreement or security tests skills).

Taking that as a starting point, SAFERtec will introduce another (i.e., a second) enhancement by providing tools and strong knowledge bases, to make the generic evaluation tasks more focused (and consequently, efficient). In fact, part of the high cost of security assurance comes from the fact that everything in the evaluation process has to be redefined and adapted to the new evaluated object. Here with our specific ITS systems scope, those objects will have if not fully standardized architecture and interfaces, at least very similar ones. Thus, efforts can be made to provide reference tests suites and tools for those recurring functions and interfaces.

The SAFERtec framework is currently under process and improvements as well as updates in the herein preliminary design should be expected. Its main objective however will remain to enhance all the design validation and tests validation phases (functional and vulnerability) by carefully adapting existing tools to the ITS environment. Finally, the SAFERtec framework is expected to be *the first* to provide security assurance arguments *at the system level*, validating not just the main system elements but the security of the complete system and its composition (of its constituting elements).

7 References

[1] NIST, *FIPS PUB 140-2 - Security Requirements for Cryptographique Modules,* 2002.

[2] SOG-IS, *ITSEC: Information Technology Security Evaluation Criteria.*

[3] ANSSI, *CSPN: Certification de sécurité de premier niveau des produits des technologies de l'information, réf. ANSSI-CSPN-CER-P-01, version 1.1,* 2014.

[4] ISO/IEC, *15408-1 Information technology – Security techniques – Evaluation criteria for IT security - Part 1: Introduction and general model,* 2009.

[5] ISO/IEC, "15408-2 Information technology – Security techniques – Evaluation criteria for IT security - Part 2: Security functional requirements.," 2008.

[6] ISO/IEC, "15408-3 Information technology – Security techniques – Evaluation criteria for IT security - Part 3: Security assurance requirements," 2008.

[7] U. D. o. defense, *Trusted Computer System Evaluation Criteria (TCSEC), DoD 5200.28-STD,* l985.

[8] B. B. K. G. a. T. W. N. Bartol, "Measuring Cyber Security and Information Assurance: a State-of-the-Art Report," *Information Assurance Technology Analysis Center (IATAC),* 2009.

[9] I. Freiling, Dependability Metrics, Springer, vol. 4909, 2008.

[10] ISO/IEC, *27004 : Information technology – Security techniques – Information security management – Measurement..*

[11] ISO/IEC, *27005: Information technology – Security techniques – Information security risk management..*

[12] R. H. a. A. S. R. Vaughn, "Information assurance measures and metrics-state of practice and proposed taxonomy," *Proceedings of Hawaii International Conference on System Sciences, vol. 1., Citeseer,* 2003.

[13] P. P. M. A. L. B. N. J. a. H. A. S. Nabil, "Current trends and advances in information assurance metricsFredericton," *Proceedings of Second Annual Conference on Privacy, Security, and Trust (PST 2004), NB, Canada,* p. 1315, 2004.

[14] A. Jaquith, Security metrics, replacing fear, uncertainty, and doubt, MA: Addison-Wesley Reading, 2007.

[15] J. P. a. J. W. M. Howard, "Measuring relative attack surfaces," *Computer Security in the 21st Century Eds. Springer US,* p. pp. 109–137., 2005.

[16] S. Payne, A guide to security metrics, SANS institute, 2001.

[17] J. McHugh, "Quality of protection: measuring the unmeasurable?," in *Proceedings of the 2nd ACM workshop on Quality of protection*, New York, NY, USA, , 2006.

[18]]. S. Bellovin, “On the brittleness of software and the infeasibility of security metrics,” 2006.

[19] ISO/IEC., “ 27001:2005 – Information technology – Security techniques – Information security management systems – Requirements.,” 2005.

[20] U. D. o. Defense, *DoD Instruction (DoDI) 8510.01, Risk Management Framework (RMF) for DoD Information Technology (IT).*

[21] C. Members, *Arrangement on the Recognition of Common Criteria Certificates in the field of Information Technology Security,* 2014.

[22] J. Officiel, *French decree n° 2002-535, Décret modifié relatif à l'évaluation et à la certification de la sécurité offerte par les produits et les systèmes des technologies de l'information,* 2002.

[23] ANSSI, *Le Référentiel général de sécurité (RGS) v2.0,* 2014.

[24] J. O. d. l. R. Française, *« Arrêté du 18 janvier 2013 modifiant l'arrêté du 8 juin 2012 relatif à la certification des équipements techniques et à l'homologation des chaînes de collecte, de contrôle automatique et de contrôle manuel de la taxe alsacienne [...]",* 2013.

[25] I. Project, *L3.1: Specification of a security evaluation process for Cooperative ITS systems,* 2017.

[26] C. m. (www.commoncriteriaportal.org), *The Common Criteria for Information Technology Security Evaluation (CC), v3.1 Release 5.*

[27] S. international, *Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems,* 2014 .

[28] SOG-IS, *Minimum Site Security Requirements version 1.1,* 2013.

[29] C. C. D. board, *Common Criteria - Supporting Document Guidance - Site Certification,* 2007 .

[30] P. Salini and S. Kanmani, “Model Oriented Security Requirements Engineering (MOSRE) framework for web applications.,” *Advances in Computing and Information Technology,* pp. 341-353, 2013.

[31] R. N. Mead and T. Stehney, “Security quality requirements engineering (SQUARE) methodology,” *ACM,* vol. 30, 2005.

[32] A. Bijwe and N. Mead, “Adapting the square process for privacy requirements engineering,” 2010.

[33] S. Miyazaki, N. Mead and J. Zhan, “Computer-aided privacy requirements elicitation technique.,” in *APSCC'08. IEEE*, 2008.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 732319

[34] C. Haley, R. Laney, J. Moett and B. Nuseibeh, "Security requirements engineering: A framework for representation and analysis," *IEEE Transactions on Software Engineering,* vol. 1, no. 34, pp. 133-153, 2008.

[35] N. Ahmed and R. Matulevicius, "A method for eliciting security requirements from the business process models," in *CAiSE (Forum/Doctoral Consortium)*, 2014.

[36] A. Van Lamsweerde, "Goal-oriented requirements engineering: A guided tour," in *Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium*, 2001.

[37] V. Lamsweerde and L. E. A., "Handling obstacles in goal-oriented requirements engineering.," *IEEE Transactions on software engineering,* vol. 26, no. 10, pp. 978-1005, 2000.

[38] S. Pachidi, " Goal-oriented requirements engineering with KAOS," 2009.

[39] A. Van Lamsweerde, "Elaborating security requirements by construction of intentional antimodels," in *26th International Conference on Software Engineering*, 2004.

[40] S. Fabender, M. Heisel and R. Meis, "Functional requirements under security pressure," in *Software Paradigm Trends (ICSOFT-PT) 9th International Conference*, 2014.

[41] S. Fabender, M. Heisel and R. Meis, " Problem-based security requirements elicitation and renement with pressure," in *International Conference on Software Technologies*, 2014.

[42] J. Castro, M. Kolp and a. J. Mylopoulos, "Towards requirements-driven information systems engineering: the Tropos project," vol. 27, no. 6, pp. 365-389, 2002.

[43] H. Mouratidis, "A natural extension of tropos methodology for modelling security," 2002.

[44] M. Deng, K. Wuyts, R. Scandariato, B. Preneel and W. Joosen, "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requirements Engineering," vol. 16, no. 1, 2011.

[45] S. Spiekermann and L. F. Cranor, "Engineering privacy," *IEEE Transactions on software engineering* , vol. 35, no. 1, p. 67–82, 2009.

[46] P. Colombo and E. Ferrari, "Towards a modeling and analysis framework for privacy-aware systems.," *In Privacy, Security, Risk and Trust (PASSAT),* pp. 81-90, 2012.

[47] A. Sabouri and K. Rannenberg, "ABC4Trust: Protecting Privacy in Identity Management by Bringing Privacy-ABCs into Real-Life," 2015.

[48] C. Kalloniatis, E. Kavakli and S. Gritzalis, "Addressing privacy requirements in system design: the PriS method," *Requirements Engineering,* vol. 13, no. 3, pp. 241-255, 2008.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

[49] J. Officiel, *French decree n° 2002-535, « Décret modifié relatif à l'évaluation et à la certification de la sécurité offerte par les produits et les systèmes des technologies de l'information »,* 2002.

[50] ECSEL Research and Innovation actions (RIA), "AMASS. Architecture-driven, Multi-concern and Seamless Assurance and. Certification of Cyber-Physical Systems. Baseline and requirements for multi-concern assurance. D4.1".

[51] E. Albrechtsen, "Security vs safety".

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319