# D1.1

# Regulative Baseline: Compositional Security Evaluation

| | |
|---|---|
| **Project number:** | 731456 |
| **Project acronym:** | certMILS |
| **Project title:** | Compositional security certification for medium to high-assurance COTS-based systems in environments with emerging threats |
| **Start date of the project:** | 1st January, 2017 |
| **Duration:** | 48 months |
| **Programme:** | H2020-DS-LEIT-2016 |

| | |
|---|---|
| **Deliverable type:** | Report |
| **Deliverable reference number:** | DS-01-731456 / D1.1/ 1.0 |
| **Work package contributing to the deliverable:** | WP1 |
| **Due date:** | June 2017 |
| **Actual submission date:** | 20th July, 2017 |

| | |
|---|---|
| **Responsible organisation:** | atsec information security GmbH |
| **Editor:** | Andreas Hohenegger |
| **Dissemination level:** | PU |
| **Revision:** | 1.0 |

| | |
|---|---|
| **Abstract:** | The regulative baseline for compositional evaluation performed using Common Criteria is documented here. It identifies the current approaches and possibilities as well as the known challenges for composition. |
| **Keywords:** | Composition, security evaluations, assurance, security dependencies, methodology |

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731456.

## Editor

Andreas Hohenegger (atsec)

## Contributors (ordered according to beneficiary numbers)

Holger Blasum, Sergey Tverdyshev (SYSGO)

Luis Garcia (E&E)

Amelia Álvarez de Sotomayor , Benito Caracuel Sillero (Schneider Electric)

Kertis Tomáš (UCO)

Andreas Hohenegger, Gerald Krummeck, Helmut Kurth, Staffan Persson (atsec)

Reinhard Hametner, Michael Paultisch, Peter Tummeltshammer (THALES)

Hager Michal (EZÚ)

## Disclaimer

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The content of this document reflects only the author`s view – the European Commission is not responsible for any use that may be made of the information it contains. The users use the information at their sole risk and liability.

# Executive Summary

In the focus of the certMILS project are cyber physical systems (CPS). These combine physical and software elements and, with the advances of industry, such automated solutions increasingly take over critical tasks in all areas of our society. Smart grids, safety-critical transport systems and in general industrial control systems – CPS can take on many faces but are commonly characterized by their complexity. CPS are composed of specialized parts and COTS elements, typically by different parties.

Due to CPS' criticality, there is a high need for assurance in the correct (safe and secure) operation of the entire systems and, consequently, they are often subject to regulations. That is, the components and/or complete systems must be certified according to standards, applicable to the respective sector. For instance, the IEC 62443 series of standards deals with complete industrial automation and control systems, the ISO 27000 series with information security management systems (processes), and the Common Criteria (CC) with subsystems of IT products.

Even though these different frameworks share some common principles, a diversity of approaches prevails. It is not always easily reconciled. The present report provides an overview of the various standards applicable to different critical applications. It points out that the regulative situation is sometimes unsatisfactorily incomplete or conflicting where different standards apply.

Prerequisite for the certification according to any standard is a successful evaluation according to its principles. The rigor of such evaluations will normally increase with increasing criticality of the application, but is in practice eventually a trade-off between assurance needs and economic feasibility. A common theme of security evaluations of CPS is therefore the desire to derive assurance for composed systems from that established for their components (subsystems).

The objective here is that the results of the component's evaluation can be reused to render the evaluation of complex systems economically feasible, or possible at all. In particular, their evaluation/certification would not need to be repeated from scratch if one or more system components are changed. To this end, one of the relevant and broadly applied industry standards, the CC knows the concept of compositional evaluation.

However, despite the rather generic formulation of this CC aspect, and the promise that it holds, it has hardly found application. The issues that hinder its success are described by the present document, as well as an alternative method, intended for smart cards and similar applications, that received more attention but likewise suffers from shortcomings. In summary, the benefit of the CC compositional evaluation approach is minor for low assurance evaluations. Very high assurance cannot be gained as it foresees a limited transfer of design documentation. This is owed to the fact that component developers will not always easily share these secrets.

MILS systems, that borrow their name to this project, arose from the requirement to gain assurance in the security properties of computers. They feature a layered structure in which security-critical functions are concentrated in a part, called the separation kernel, which is intentionally small enough to permit evaluation with great rigour. In applications, such as CPS, these layers are always combined with other elements, such as the hardware platform or software running on top of the separation kernel.

At first glance, MILS systems seem to lend themselves to compositional evaluation, as they are well structured and characterized by strong security policies. However, the various conceivable applications of compositional evaluation suggested by MILS applications still pose challenges for the existing methods if high assurance is required. It is the purpose of the present report to describe the different approaches and what they have to offer for this type of system.

# Contents

# List of Figures

# Chapter 1 Introduction to Composition

IT products and systems are usually not monolithic. Rather, they consist of various components, such as software and hardware parts. The security features of most modern products are established by the co-working of such components. Since these are typically contributed by different vendors, they are often evaluated independently from each other. This means that the confidence in the security of IT products or systems, quite often, rests on separate evaluations of their components. Each of these independent assessments has a limited scope and will normally make general assumptions about other components that are considered part of the environment.

Consequently, to gain confidence in the product's security, it must be assumed that the mutual dependencies of the different parts are in fact satisfied, such that the security policy of the whole product is met. While this may be sufficient for low assurance, it is not for medium and high assurance. In the alleged unsafe environments the attack potential is expected to be higher and dependencies are far more critical. The interactions of the components may affect security properties like separation and possibly introduce side channels.

It is quite obvious that these critical dependencies cannot be addressed solely by the isolated evaluation of the components, each making assumptions about the others. In theory, it is possible to perform a complete evaluation of all components together. However, this is for most IT products neither technically nor economically feasible, due to limited access to developer information, time or resources. In monolithic evaluations the effort grows drastically with the complexity and size of the evaluation target. To keep pace with growing system complexity, assurance needs and development speed it is not a future-oriented approach.

The concept of considering multiple evaluated components within a new evaluation is called composition. The idea is to perform the evaluation with significantly less effort, as compared to a monolithic evaluation, and nevertheless to provide a high level of assurance. With this prospect, a compositional evaluation approach appears to be a promising strategy.

As will be discussed, there are many different variants of composition, but the general problem is unsolved. More accurately, it is possible to solve it at low-assurance, but more or less impossible at high assurance. However, by simplifying the problem, restricting it to basic scenarios, each with its specific conditions and limitations, it is not only possible to understand it, but also to address it successfully.

The Multiple Independent Levels of Security (MILS) approach to security is based on the idea of building isolated security critical functionality into components that are as small and simple as possible, and whose security policies are likewise elementary. The different components of a MILS system are separated and information flow between them takes place in a well defined way. This means that the MILS architecture facilitates the assessment of the security properties of IT products, suggesting composition as the strategy for their evaluation.

This report, the regulative baseline, identifies different existing approaches to composition evaluation, discusses how they can be applied and what their limitations are. It is based on previous work performed in the EURO-MILS project [1] and others.

# Chapter 2 Common Criteria and TOE Scope

The Common Criteria (CC) [2] are widely used to gain assurance in the security of software, firmware and/or hardware components. Within the framework of the CC, evaluations are performed for these components and the accompanying guidance documents. The scope of what exactly is subject to the evaluation is determined by the developer or sponsor. It is not tied to the boundaries of the IT products as commonly understood. For this defined scope of evaluation the CC introduces the term “Target of Evaluation” (TOE).

Usually, the TOE is only a part of an IT product while the rest of it, including other important components, is treated as environment. To counter the threats and to meet its security objectives the TOE may make certain (reasonable) assumptions about that environment.

These assumptions, and the security objectives for the operational environment meant to uphold them, are not subject to the evaluation (they are not considered in the assessment of the design documentation and so on) and therefore not really verified. Treating large parts of an IT product as environment, or making strong assumptions about it, can thereby severely simplify the evaluation.

For example, software evaluations usually assume that the underlying hardware has certain security properties, although the hardware itself is outside of the scope. Still, the independent evaluation of the software creates some amount of confidence in the security of an IT product that end users may rely on when they choose their vendor.

If the security functions of the whole product are provided by the co-working of several of its components, the assurance gained in the evaluation of one or multiple component TOEs may however be insufficient. Especially, since the evaluations are not automatically consistent with respect to their evaluation assurance levels (EALs) or the assumptions made about their respective operational environments. Even if all components have been TOEs in separate successful CC evaluations, it is by no means excluded that these parts interfere in a destructive way, evading the security policy of the whole product.

As pointed out, the most obvious approach, to examine the combined IT product as the TOE within a monolithic evaluation, is often not feasible for various reasons. Instead of starting from scratch, one might therefore want to benefit from the assurance gained in the CC evaluations of the individual components. Such a composition approach needs to specify what additional evaluation efforts are necessary to establish assurance at the desired level.

# Chapter 3 Types of Composition and Problems

Composition is an important concept for security evaluations, because it promises to allow cost efficient evaluations of relatively complex products by re-use of the results from the evaluation of its components. While this is quite obviously difficult to achieve in general, there are simple special cases in which the evaluation of the whole IT product benefits from the previous evaluation of a single component.

These differ in how the considered component relates to the whole product. Three simple types of composition may be distinguished to work out the challenges entailed by different product designs:

- Layering composition,
- Network composition,
- Component composition.

These types vary in the inter-dependence of the different components and, correspondingly, in the way in which they would have to be treated in the composed product evaluation.

Note that the evaluation of the composed product not only depends on the type of composition, but also on the security policy used in the assessment of the components and the assurance level as well as the overall policy of the composed product. All these aspects must be taken into account when performing such compositional CC evaluations.

## 3.1 Layering composition

Layering, as a composition type, describes the scenario where one component is built on top of another one (see Figure 1).

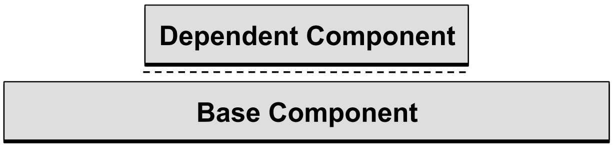

Figure 1: Layering composition.

Examples for layered composition are

- hardware – software layer,
- operating system – application layer,
- smartcard hardware platform – smartcard application.

This scenario is reminiscent of a composition approach applicable to smart cards (to be discussed in Section 4.2) where the lower component is assumed to be certified and called “platform” and the higher component is called “application”. The evaluation of the composed TOE can in this case benefit from the evaluation of the platform which may be the hardware part of a smart card.

Simplifying assumptions that can be made for this type of composition are as follows:

- The lower component is independent of the higher one.
- The lower component is not modified by the higher one.
- The higher component uses the functions of the lower one, and *not* vice versa.

These assumptions may be key to an efficient composite evaluation, because only then the already evaluated lower base component does not need to be re-examined.

Still, a layered composition poses some specific problems. In any case, the higher component will use some security functions of the lower component. The composite evaluation must assess how the security functions of the higher component make use of those of the lower component. Potential problems here are:

- Security properties (like information flow control, fault tolerance, separation) claimed in the composite evaluation might be subverted by unspecified side effects of the lower component. It needs to be assessed if such side effects exist and if they have been considered in the lower component's evaluation.
- Alternate access paths or communication channels may exist in the lower component, possibly circumventing the higher component's security functions. They may have not been considered in the lower component's evaluation.
- Security functions of the higher component may depend on functions of the lower component which have not been considered in the lower component's evaluation. An example could be an operating system's file locking mechanism used by a database. Another critical factor to consider here is propagation of errors from one component to the other.

## 3.2 Network composition

In a network type composition (compare Figure 2), one component (B) uses specific functions of another component (A), requesting these security functions over a (usually protected) communication channel.

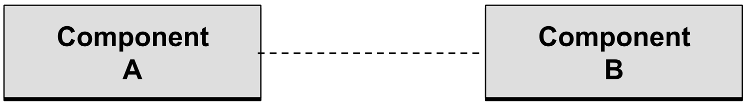

Figure 2: Network composition.

An example for this type of composition would be a product using the functions of an external LDAP server. For instance, component B might use component A as its user database holding the identification and authentication (I&A) information.

Another example could be a client-server scenario where the client (component B) does not perform the I&A mechanism, but would forward a login request of a user to the server (component A), which would validate the access credentials and tell the client if the user was allowed to login on the client. The client would then only enforce the server's decision.

In this scenario, there is clearly some influence between both components. For example, the request of the client might trigger an update of the failed login count kept by the server. To make such an evaluation feasible, the following assumptions should hold:

- The security inter-dependencies are clearly described.
- Both products are separated such that there is no channel or influence other than the defined one.
- Both products implement the functions required to protect the communication channel.

The specific issues to be clarified in a composite evaluation of such an architecture are somewhat similar to those of the layering composition:

- Security functions may not fit together.
- Assumptions made about one of the components may be invalid; as an example, critical data transferred to the other component may not be sufficiently protected.

- Security functions may induce unwanted side effects, like covert channels leaking cryptographic keys.

## 3.3 Component composition

In a component type composition, one component appears as a part of a larger component (compare Figure 3).

A typical example for this type of composition would be a library or subsystem providing a specific security function as part of a larger product. For instance, a crypto library that is used by many different products. The crypto library has been evaluated and that this evaluation result should be re-useable in products using it.

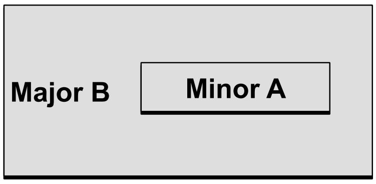

Figure 3: Component composition.

This type of composition is more complex to assess than the other types, because usually there will be no clear separation between the parts. Therefore, each component may influence the other via channels that differ from the intended ones.

This produces a number of issues that must be assessed during the composite evaluation:

- Security functions may be bypassed due to the lack of sufficient separation.
- One component may modify security functions and policies of other components and therefore for the whole product.
- Due to the lack of separation, all sorts of critical side effects may occur.

# Chapter 4 Existing Approaches

Beyond the generic classification of composition types, that largely serves as an illustration, there are two well-described frameworks for composition evaluations:

- The CAP approach, as specified in CC Part 3 [3].
- Composition for smart cards, as specified in “Composite product evaluation for Smart Cards and similar devices” (CCDB) [4].

In brief, the CAP approach described in CC v3.1 provides a generic, low-assurance method. It seems mostly suited to applications on top of an evaluated platform (layering type of composition, Section 3.1) but is intended to be applicable for use cases of different type.

On the other hand, CCDB has been developed for smart cards and their high assurance requirements. It is suited to applications based on a smart card platform and not as generic as the ACO approach. Here, the composition is akin the layering type, but some amount of back reaction on the platform is allowed.

Since the former is limited to attack potentials of up-to “Enhanced-basic”, none of these can be considered a generic solution for high assurance and all types of composition. For very low assurance level, such as EAL1 or EAL2, evaluations are possibly better performed as new evaluations, since there is limited or no benefit in reusing the results of component evaluations.

## 4.1 The CC composition approach – CAP

Part 3 of the Common Criteria [3] introduces an assurance class (ACO) dedicated to the evaluation of composed IT products. This addition is motivated by the poor scaling of the evaluation of complex targets [5] and the limited availability of developer evidence. The level of assurance gained for the composed TOEs considered in such evaluations expresses itself in the choice of so-called “Composed Assurance Packages” (CAP-A, CAP-B, and CAP-C). These are not directly comparable to the Evaluation Assurance Levels (EAL1-7) common to regular CC evaluations. Therefore, for composed products, the CC partly looses its function as a standard ruler for security assessments.

The underlying scenario is mostly one of layering type (compare Section 3.1), where a “dependent component” uses security functions of a “base component” (as emphasized in [3] more difficult dependencies may also be considered, e.g. through an iterative treatment). The ACO class (Assurance class Composition) assumes that both these components were TOEs in separate successful CC evaluations.

The five families of assurance requirements in the ACO class have to be selected for composed TOE evaluations and serve to analyse the interaction between the parts of the composed TOE:

- In ACO_REL (Reliance of dependent component) the dependencies of the security functions on the base component are identified in terms of its interfaces.
- The identified interfaces (and possibly supporting ones) are analysed in ACO_DEV (Development evidence).
- ACO_COR (Composition rationale) demands a composition rationale which argues that for the base component appropriate assurance requirements have been used and that it is used in the evaluated configuration.
- ACO_CTT (Composed TOE testing) and ACO_VUL (Composition vulnerability analysis) lay out steps for the testing and vulnerability analysis of the composed TOE. These families use incomplete specifications, possibly identified in ACO_COR as input.

As compared to a monolithic CC evaluation of the composed TOE, the CC composition approach focuses on the security of the interfaces. Hence, the potential benefit of a composed evaluation based on the ACO class is that the composed TSF does not need to be re-evaluated. While full developer evidence is assumed to be available for the dependent component, the base component’s evaluation results are reused and the evaluation of the composed TOE is largely independent of such documents. This implies that no additional work is needed on the side of the developer of the base component.

However, there are hassles, beyond the mere extra effort required for the analysis of the interfaces:

- The required EALs of the component evaluations increases with increasing CAP-level (e.g. EAL4 for each component in a CAP-C composed evaluation).
- A basic EAL1-evaluation of the composed system may be required (at CAP-C).
- Since the ACO approach assumes that all components were previously subjects to separate CC evaluations, it does not allow any significant additions of software or hardware that might be required to mediate the interaction between the parts.
- For composite TOEs with more than two components, the analysis of the interfaces needs to be iterated for each pair of base and dependent components.
- The vulnerability analysis is hindered by the limited depth of available design documentation for the base component. The highest attack potential that can be expressed with CAPs is therefore “enhanced-basic”.

The gained confidence at highest composite level, CAP-C, is comparable to that gained in a monolithic EAL4 evaluation [1]. This degree of assurance may therefore be insufficient for safety-critical systems for which higher levels may be required. Hence, the applicability of the CAP approach is limited. On the other hand, at low EAL, a monolithic evaluation of the composed TOE may add only little overhead as compared to a composed evaluation (assuming that sufficient documentation is available for the independent component).

## 4.2 The CC composite approach – smart cards

The concept of compositional evaluation is as well relevant to smart cards. These consist of software applications that run on top of the smart card hardware. Often these two components are produced by different vendors. However, the assurance needs for smart cards are well beyond CAP-C, the highest level that can be reached in an evaluation based on the ACO class.

The “Common Criteria Development Board” (CCDB) has therefore issued the “Common Criteria Supporting Document Composite Product Evaluation for Smart Cards and Similar Devices” (CPE) [4]. It applies to IT products whose architecture is reminiscent of the layering type composition discussed in Section 3.1. The result of a successful evaluation according to these guidelines may be a certified product associated with an EAL level suitable for comparison with regular CC evaluations of monolithic TOEs.

In CPE jargon, the base- and dependent components are called “platform” and “application” respectively. However, in contrast to the assumptions listed in Section 3.1 for layering type composition, the CPE allows for a back-reaction of the application on the platform. This is owed to the fact that, to provide the security services of the overall product, the application often activates, configures or manages the security functions of the hardware platform. Furthermore, the CPE permits an arbitrary amount of additional “wiring” that is part of neither the application nor the platform. This is in contrast to CAP evaluations in which it is assumed that the dependent component only uses security functions provided by the base component through its interfaces, without modifying it. In view of these differences the logical structure of this type of composed TOE is better displayed as in Figure 4.

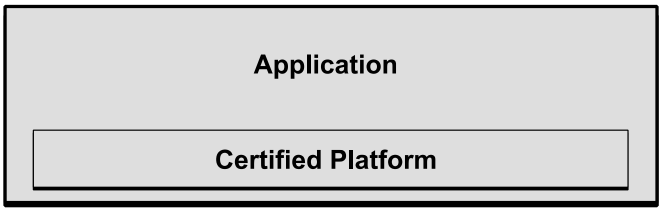

Figure 4: Smart card (CPE) composition.

This representation also better accounts for the evaluation strategy underlying CPE. It assumes that the platform is already CC certified at a sufficiently high assurance level. The developer of the application integrates this platform in his application and performs an evaluation of composed TOE, using information that is made available by the platform developer.

These differences make the CPE approach the more flexible and powerful one. Eventually, in a CAP based evaluation the development, and perhaps the evaluation, of the components is supposed to be ready at the time of integration. Only then is their mutual interaction analyzed. It is by no means limited to the smart card domain and has actually been applied in a number of cases.

In contrast to CAP evaluations, CPE does not introduce a new class of assurance components. Instead, it complements the existing CC classes with additional assurance families that carry the postfix _COMP. Apart from these additional assurance families, the CPE framework includes the corresponding evaluation methodology extending the "Common Methodology for Information Technology Security Evaluation" (CEM) [6] which complements the Common Criteria.

More specifically, CPE augments the CC catalogue with the following assurance families:

- In ASE_COMP it is assessed that the security target of the composed TOE is consistent with that of the platform, mainly with respect to the security problem and the objectives.
- ALC_COMP determines the compatibility of the delivery and acceptance procedures of the platform developer and the composed product integrator.
- The compliance of the composite product with the requirements on the platform environment is investigated in ADV_COMP.
- ATE_COMP specifies the functional testing of the integrated TOE, comprising platform and application, with respect to its security target.
- AVA_COMP determines vulnerabilities of the composed TOE in its target environment.

The evaluation of these aspects makes use of the results of the upstream certification of the platform component. This information is supposed to be available early in the development and evaluation of the enclosing application. Its availability can render a CPE evaluation economic compared to the monolithic evaluation of the same TOE.

Like the CAP approach based on the ACO class, the CPE approach for smart cards and similar devices does not come without its down sides:

- The evaluation does not benefit from possibly available evaluations of the application or a part of it (dependent component).
- The effort for re-evaluations/certifications with a modified component may be high since the full vulnerability analysis may need to be repeated.
- At very high assurance level there is a risk that vulnerabilities, relevant at this attack potential, are overlooked in an evaluation because of limited information flow between the developer of the base component and that of the composed TOE (ETR for composite evaluation, or ETR-lite) [7].

## 4.3 Related considerations

### *4.3.1 Non-interfering composed evaluations*

As part of the EURO-MILS project [1], shortcomings of existing compositional evaluation methods were recognized. In particular, the problem that within the ACO methodology the vulnerability analysis has to repeated for the composed TOE, at a level of rigor commensurable to the required CAP level. This is necessary to diminish the risk of unintended interactions (dubbed interference) between components, given that the integrator (developer of the composed TOE) has only high-level design documentation at his hands.

As part of the EURO-MILS project a whitepaper has been published [8], which outlines a method called "Non-Interfering Composed Evaluation". This method seeks to address the issue of unwanted interference by requiring that its absence be demonstrated evidently during the certification of the component TOEs. Once this has been demonstrated for all components, the evaluation of the composed TOE can be focused on the functional (intended) interactions between these parts.

As the authors assert, the method shifts some of the evaluation burden from the developer of the composed TOE to the evaluators (developers) of the components. Latter needs to demonstrate the "non-interference property". Following, the authors, he does that by accurately describing all explicit and implicit interfaces. Indeed, it remains to be exemplified that this property can actually be shown for a component. In the absence of detailed knowledge about the possible application it seems hard to prove – in full generality – that there will not be any unintended side channels.

### 4.3.2 IEC 62443 and relations to CC artefacts

The Common Criteria represent an established approach for the evaluation of the security of (parts of) IT products. While the CC framework is flexible and explicitly does not preclude its application in different areas, it is at present not used for the assessment of larger systems such as Industrial Automation and Control Systems (IACS). However, like other IT security standards, it shares some ideas with frameworks tailored to this kind of problem.

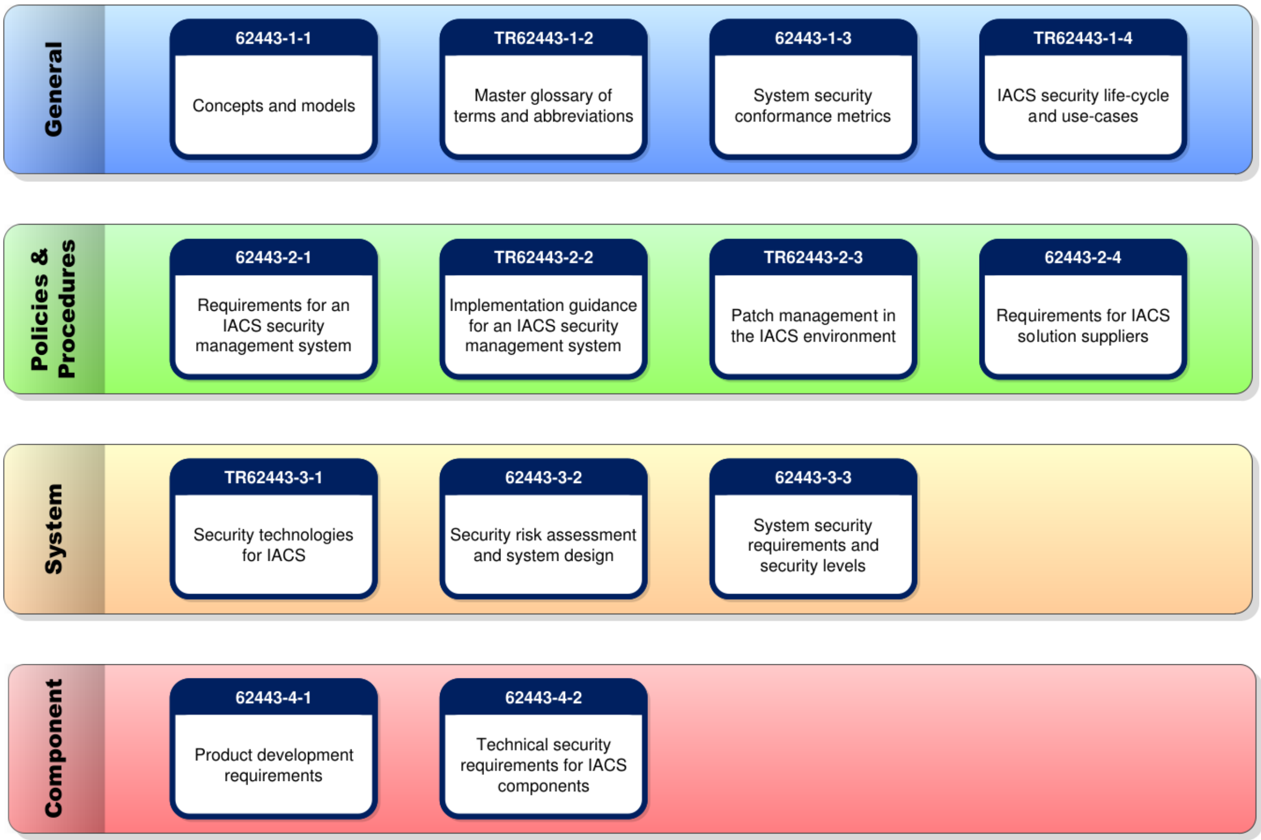
General
62443-1-1
Concepts and models
TR62443-1-2
Master glossary of terms and abbreviations
62443-1-3
System security conformance metrics
TR62443-1-4
IACS security life-cycle and use-cases
Policies & Procedures
62443-2-1
Requirements for an IACS security management system
TR62443-2-2
Implementation guidance for an IACS security management system
TR62443-2-3
Patch management in the IACS environment
62443-2-4
Requirements for IACS solution suppliers
System
TR62443-3-1
Security technologies for IACS
62443-3-2
Security risk assessment and system design
62443-3-3
System security requirements and security levels
Component
62443-4-1
Product development requirements
62443-4-2
Technical security requirements for IACS components

Figure 5: The elements of the IEC 62443 series. [Image: WikiMedia, CC BY-SA 3.0].

ISA, the Instrumentation Systems and Automation Society, together with IEC, is the developer of the IEC 62443 standard [9], which targets IACS. Its structure reflects the compositional architecture of industrial plants. See Figure 5 for an overview. As examples for IACS the first part of the IEC 62443 series mentions "control systems used in manufacturing and processing plants and facilities, building environmental control systems, geographically dispersed operations such as utilities (i.e., electricity, gas, and water), pipelines and petroleum production and distribution facilities, and other industries and applications such as transportation networks, that use automated or remotely controlled or monitored assets". The parts IEC 62443-3-1 to IEC 62443-3-3 target entire plants, and parts IEC 62443-4-1 and IEC 62443-4-2 cover individual components. IEC 62443-4-1 and IEC 62443-4-2 are still under development. However, ISA has already proposed and applied a certification scheme according to IEC 62443-4-1 and IEC 62443-4-2:

- SDLA for based on IEC 62443-4-1 [10],

- EDSA based on IEC 62443-4-2 [11].

In the sections below, some relationships that exist between aspects of IEC 62443 and the Common Criteria are pointed out. It is emphasized that these considerations merely represent possible starting points for further analysis required to benefit from the combination of these two different frameworks.

### 4.3.2.1 IEC 62443 certification and CC similarities

The intention of the IEC 62443 standard series [10] is to build extensions to enterprise security that adapt requirements for IT management systems and combine them with the unique requirements that embrace the strong availability needed by IACS.

IEC 62443 provides a flexible framework to evaluate the security functions of such systems. This framework encompasses seven foundational requirement (FR) groups (IEC 62443-3-3) detailed below. These FRs are closely related to the security functional requirements (SFRs) defined in CC Part 2. That is, CC SFRs could in principle  be used to complement the FRs of IEC 62443-3-3.

IEC 62443 also defines requirements for security policies, procedures, and practices applicable to IACS solutions throughout their life cycle, describing what shall or should be provided during integration and maintenance activities (IEC 62443-2-4). Within the CC, the ALC class (live-cycle support) takes a similar point of view, rendering it a useful source to clarify or extend the IEC 62443 requirements.

There are already certified products meeting these IEC 62443 requirements, such as the station automation system SICAM PAS/PQS and SICAM AK3 by Siemens [12].

#### 4.3.2.1.1 Framework

IEC 62443-3-3 provides detailed technical system requirements associated with seven foundational requirements:

- Access control,
- Use control,
- Data integrity,
- Data confidentiality,
- Restrict data flow,
- Timely response to event,
- Resource availability.

These requirements can be used by the IACS community when developing the appropriate technical system target Security Assurance Level (SAL) for specific assets.

#### 4.3.2.1.2 Integration and Maintenance

Requirements for the security capabilities of providers of integration and maintenance services for IACS are specified in IEC 62443-2-4.

An example for such a capability requirement is: “The Supplier shall ensure that the point of connection with the Solution’s control system network to the SIS is protected with a network security device with documented and maintained security rules. The intent of this requirement is to ensure that the Solution provides logical or physical separation of SIS/safety-related communications (for SIL 1 and above) from communications on other Solution networks.”

All capability requirements are denoted in a systematic way, according to the following scheme:

- Sec Req ID: The “Security Requirement ID” consists of three parts separated by dots “.”. The first part is “SR”, indicating a security requirement. The second part is a unique functional area ID which refers to one of
  - Solution staffing,
  - Security incidents,

- Security tools and evaluations,
- Architecture,
- SIS,
- Wireless,
- Account management,
- Malware protection,
- Backup/Restore,
- Patch management.

The third part is a numerical identifier for the requirement, assigned within the functional area.

- BR/RE: Indicates whether the requirement is a base requirement or a requirement enhancement.
- Solution integration: Indicates that the requirement applies to integration suppliers.
- Solution maintenance: Indicates that the requirement applies to maintenance suppliers.
- Functional area: Provides the top level organization of the requirements.
- Base requirement: Contains the keyword that best describes the base requirement.
- Detailed requirement: Contains the keyword that best describes the detailed requirement.
- Requirement enhancement: Contains the keyword that best describes a requirement enhancement. Requirement enhancements are defined as extensions/specializations of base requirements.
- Requirement description: Contains the textual description of the requirement. It may also contain notes that are examples provided to help in understanding the requirement.
- Profiles: The concept of profiles can be used to enhance this specification for a particular industry or industry sector. The specification defines three default profiles:
- Base Profile (Base requirements): Requirements identified using the BR in the BR/RE column.
- Enhanced Profile #1 (Enhancements of base requirements): Requirements identified using the RE (#) format in the BR/RE column.
- Enhanced Profile #2 (Enhancements of base requirement enhancements): Requirements identified using the RE (#A) format.

As an example, consider the classification of the capability requirement quoted above:

| | |
|---|---|
| Sec Rq ID | SR.05.01 |
| BR/RE | BR |
| Solution integration | Yes |
| Solution maintenance | Yes |
| Functional area | SIS |
| Base requirement | Network interfaces |
| Detailed requirement | Control system networks |
| Requirement enhancement | None |

| | |
|---|---|
| Requirement description | The supplier shall ensure that the point of connection within the Solution´s control system network to the SIS is protected with a network security device with documented and maintained security rules. The intent of this requirement is to ensure that the Solution provides logical or physical separation of SIS/safety-related communications (for SIL 1 and above) form communications on other Solution networks. |
| Profile | BP |

#### 4.3.2.2 Compositional crosslink for operating systems from CC to ISASecure SDLA interpretation based on IEC 62443

There is an interesting compositional cross-link from IEC 62443 to the CC as follows. For compositional certification, starting from ISASecure/SIL2, SDLA-312’s [13] MIV-5 (Module Implementation & Verification) work unit states that: “If the product includes a Commercial off the Shelf (COTS) operating system, then the operating system shall either meet the requirements of this development phase or be certified to Common Criteria EAL 3 or higher or be certified to a comparable security standard, or compensating controls must be included in the product to ensure that security vulnerabilities in the operating system do not result in vulnerabilities above a certain severity level in the product.”. ISASecure is discussed in the context of the smart grids, see Section 5.1.2.1 below.

#### 4.3.2.3 Mapping aspects of CC protection profiles to IEC 62443

An example of security functions and component requirements mapping of a PLC protection profile security objectives to IEC 62443-4-2 functional requirements is given in [14] (Section 8.2.4). PLCs (Programmable Logic Controllers) are a component frequently employed in IACS.

# Chapter 5 Areas of Application

In the focus of this report are cyber physical systems (CPS). They often feature a substructure on several levels (hierarchies) and are subject to regulations. That is, the components and/or entire systems must be designed and evaluated/certified according to standards applicable to the respective sector. As the subsequent sections, contributed by certMILS partners operating in the respective fields, show the regulatory situation is usually diverse. Different aspects are addressed by different standards, while others are not addressed at all or covered by more than one possible approach.

Clearly, CPS are candidates for the application of compositional evaluation strategies if they exhibit a layered or networking structure. If components can be identified which are well separated, interacting dominantly through well defined channels, such methods may allow to draw conclusions about the security of the whole system from the results of the evaluation of their parts. This holds true also for the application of a separation kernel for which possible use cases are described in a dedicated subsection.

## 5.1 Smart grids

The concept of smart grids has emerged with the modernization of electric grids, integrating information technologies at different points of the power system, connecting generators and consumers, this way, evolving them from one-way/energy-only grids into two-way energy&data smart grids. Thus, the smart grid combines electric networks and IT infrastructure to integrate and interconnect all participants in order to efficiently balance demand and supply in increasingly complex networks.

Another key characteristic of the smart grid is that it brings together many different players, some of them established, others quite new, such as distributed renewable energy providers, active end-users or energy traders.

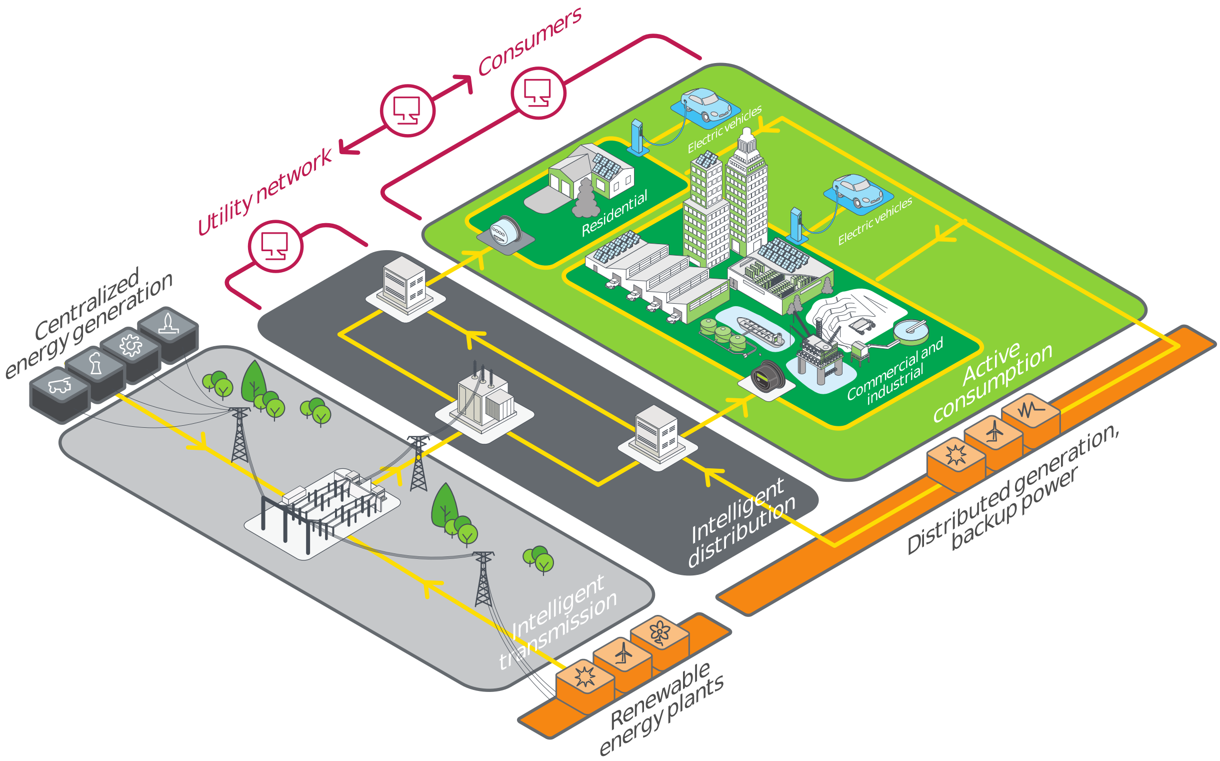

Figure 6: Components of smart grids and their interactions [Image: Schneider Electric].

As mentioned by the European Technology Platform Smart Grid [15], it employs innovative products and services together with intelligent monitoring, control, communication, and self-healing technologies to

- Better facilitate the connection and operation of generators of all sizes and technologies;
- Allow consumers to play a part in optimizing the operation of the system;
- Provide consumers with more information and greater choice of supply;
- Significantly reduce the environmental impact of the whole electricity supply system;
- Maintain or even improve the existing high levels of system reliability, quality and security of supply;
- Maintain and improve the existing services efficiently;
- Foster market integration towards an integrated European market.

### *5.1.1 Need for security in smart grids*

Electric grids are considered critical infrastructure as they are essential for the well-being of the citizens. Attacks on this infrastructure are not limited to office computers and networks. They can also directly impact the citizen's lives and economy as a whole. For instance, in December 2015, Ukraine's electric grid got hacked. The attackers managed to remotely access and manipulate the grid's industrial control system by stealing user credentials. Because of this attack, over 225.000 customers experienced a service outage for three hours. [16]

The deployment of smart grids, and the implied replacement of manual operation with automatic solutions, implies that the reliability of the power system has become strongly linked to the reliability of the IT infrastructure used to exchange monitoring signals and commands entailed by automation algorithms. Another consequence of the deployment of smart grids is that the power system operation becomes increasingly complex, making failures or mistakes more likely and growing impact in scope and cost.

In this new paradigm, communication protocols are considered one of the most critical elements for the reliable operation of the power systems, as they form the basis for the exchange of monitoring information and control commands between the different actors.

All these reasons reinforce the need for cyber security in the smart grid domain, in order to protect all relevant assets from hazards such as deliberate attacks, operation mistakes, malfunction of equipment, communication failures, natural disasters, or other events that could have fatal impact on the reliability of the energy supply.

To summarize, cyber security applied to smart grids needs to address:

- Increasingly complex architectures with a large number of elements, interfaces and communication channels, provided with different levels of protection in the underlying system.
- Increasingly complex devices such as Advanced Remote Terminal Units (RTU), Bay Controller Units (BCU) and Intelligent Electronic Devices (IED) that are evolving into smart devices and use serial and Ethernet communication, data logging, analog and digital input/output, etc. New vulnerabilities and threats have emerged to these types of assets, so their protection against cyber-attacks is a primary concern.
- Use of Smart Metering systems located in private households making the privacy of processed information critical for both, users and utilities.
- Integration of legacy components stemming from traditional electric grids. These components are usually not equipped with modern security mechanisms, as they were developed and installed when security was not a primary concern.
- Most of the connections between the various components of the smart grids are based on Internet technologies and IP-networks. This kind of infrastructure is highly vulnerable to already existing mal-ware and its capacity to spread quickly.

### *5.1.2 Standards and certification*

While the importance of applying security mechanisms to smart grid utilities has been recognized, service provider and equipment manufacturers face the following problems when trying to certify the security functions of smart grid components.

- Lack of Consensus: As we will see later, there are different approaches dealing with the cyber security of smart grids entailing a lack of uniform criteria.
- Long certification period: Due to the complexity of the smart grid infrastructure, the certification process normally takes more time than that needed for new threats to appear.
- Heterogeneous infrastructures: There is not a common infrastructure for smart grids that can be used as a reference model to create a certification methodology to be followed step-by-step. Each smart grid encompasses a large number and variety of components, communication interfaces, legacy equipment whose security properties have not been assessed, as well as new technologies installed in response to feature requests.
- Common Criteria: in order to be applicable to the smart grid domain they should be complemented by specific protection profiles.
- Certification conditions: Currently, evaluations are performed in laboratories where real operation conditions are difficult to reproduce. Therefore, the behaviour of certified components, when configured and integrated in the whole system, operating under real conditions, may be very different from the certified conditions.

Before stepping into details of the different certification schemes, we provide a list of the most significant standards that are currently used, or may be considered to be used, to support the security certification of smart grids:

- ISO/IEC 27001 and ISO/IEC 27019: ISO/IEC 27001 [17] is a standard that provides requirements for information security management systems. In the context of smart grids, the requirements collected in this standard can be used to certify policies and procedures of an organization for developing, producing, building or operating smart grids and/or components thereof.

  In addition, based on ISO/IEC 27002, ISO/IEC 27019 [18] provides guidelines for information security management in process control systems. Those principles could be applicable to the process control systems deployed for smart grids. However, these standards are merely informal and not relevant for certification as ISO/IEC 27001 requirements are.

- IEC 62443: The IEC 62443 series of standards, which evolved from ISA99, is built on established standards for the security of general purpose IT systems but identifies and addresses the differences and peculiarities of Industrial Automation and Control Systems (IACS).

  The 62443 series focuses on the functional security of IACS systems and their components to improve their safety, availability, integrity and confidentiality. This implies that the IEC 62443 does not provide specific details about the technical implementation of a secure IACS, but the  requirements regarding its functionality and/or behavior.

- ISO/IEC 15408 (Common Criteria): The CC provides a standard framework for the evaluation of security aspects of IT products or systems which has been adopted by the ISO [19]. With this target, the standard collects both, security functional and security assurance requirements (compare Section Chapter 2).

  It integrates security criteria and arguments aimed at different roles concerned with such technologies: manufacturers, evaluation laboratories and users. In the smart grid domain, it is used to verify if a product meets the needs of a customer regarding the technical implementation of a set of security functions.

- IEC 62351 (Power systems management and associated information exchange - Data and communications security): As can be deduced from its title, this set of standards focuses on the security of information exchanged to perform operations on power systems. It was developed to evaluate the security of the communication protocols defined by the IEC TC 57, including the IEC 61850, IEC 60870-5, IEC 60870-6, IEC 61970 and IEC 61968 series.

  Among the security aspects addressed by IEC 62351 are information exchange authentication with digital signature, listening prevention, identity theft prevention and intrusion detection.

- IEEE 1686 (IEEE Standard for Intelligent Electronic Devices Cyber Security Capabilities): The scope of the IEEE 1686 standard [20] is to define functions and features that should be provided in IEDs to suit the protection program for critical infrastructure.

  It describes security functions such as access, operation, configuration, firmware revision and data retrieval from the IED. The standard does not address aspects related to communication of power system protection, such as encryption of data within and outside of the substation.

In the subsequent, the certification processes related to smart grid security are summarized:

#### 5.1.2.1 ISASecure

The “ISA Security Compliance Institute” (ISCI) is a non-profit consortium of the automation controls industry that manages ISA’s automation compliance certification program [21].

The ISCI does not operate an internal testing lab but, instead, qualifies partner labs to perform IACS cybersecurity assessments.

ISASecure certification labs are independently accredited by ISO/IEC 17011 accreditation bodies to the ISO/IEC 17065 (international standard for certification bodies) and ISO/IEC 17025 (International standard for test laboratories).

ISASecure issues three types of certificates:

- Embedded Device Security Assurance (EDSA): It focuses on the security of embedded devices and addresses device characteristics and supplier development practices. It certifies the conformance of IACS components with the international standards IEC 62443-4-1 (Product Development Requirements) and IEC 62443-4-2 (Technical Security Requirements). Both parts of the standard are currently in draft status.

EDSA certification differentiates between three increasing levels of device security assurance. The achieved level depends on the following aspects: “Functional Security Assessment” (FSA), “Software Development Security Assessment” (SDSA) and “Communication Robustness Testing” (CRT).

- System Security Assurance (SSA): SSA certification is linked to SDLA certification (see below), given that it is a prerequisite to have passed a security development lifecycle process evaluation. For this reason many suppliers apply for SSA and SDLA certification in parallel.

  SSA certification of systems integrates four elements: “Security development artifacts for systems” (SDA-S), “Functional security assessment for systems” (FSA-S), “Functional security assessment for embedded devices” (FSA-E) and “System robustness testing” (SRT).

  SDA-S analyzes the artifacts generated by supplier’s development processes applied to the system to be certified. FSA-S evaluates the security capabilities of the system. FSA-E examines the security capabilities of any embedded devices integrated in the system, recognizing that in some cases security functionality is provided by other system components. SRT has three major elements – “Vulnerability Identification Testing” (VIT), CRT and “Network Stress Testing” (NST).

  In terms of standard compliance certification, the SSA FSA-S requirements for certification include all requirements of IEC 62443-3-3 (Security for industrial automation and control systems – System security requirements and security levels).

- Security Development Lifecycle Assurance (SDLA): The SDLA program certifies the development lifecycle processes of a supplier’s product becoming part of an IACS.

  As for EDSA, SDLA also integrates four ascending certification levels of development lifecycle security assurance. In the application process it needs to be specified whether certification is aimed at the development of components, systems or both, as well as the scope of the product to which the organization applies the process.

  ISASecure SDLA process certification is being aligned with the requirements and leveling concepts introduced in IEC 62443-4-1 (Security for industrial automation and control systems – Product development requirements). As mentioned previously, this section of the standard is still under development.

#### 5.1.2.2 ISO Certification (including CC)

The International Organization for Standardization (ISO) [22] does not perform certifications, and so does not issue certificates. However, ISO’s “Committee on Conformity Assessment” (CASCO) has produced several standards related to the certification process.

When selecting the certification body for a product, system or procedure, it is important to check if it uses the relevant CASCO standard and whether they are conveniently accredited.

Common Criteria Certification: The “Common Criteria Recognition Arrangement” (CCRA) is an international agreement which ensures that products can be evaluated internationally by competent and independent licensed laboratories to determine the fulfilment of a specific set of security properties, to a certain extent or assurance. In Spain, Epoche and Espri is one of these licensed laboratories. The certification of the security properties of an evaluated product can be issued by several Certificate Authorizing Schemes. The National Cryptologic Center (CCN) accredits Common Criteria Testing Laboratories operating in the Spanish scheme. Atsec Information Security is a CC evaluation facility accredited by the German, Swedish and Italian schemes. A full list of national schemes can be found online [23].

#### 5.1.2.3 Achilles Certification

The Achilles certification process is based on a set of unique network test platforms that are specifically designed to allow equipment manufacturers to carry out exhaustive network security and robustness testing throughout the product development lifecycle.

In particular, it provides automatic black box tests to evaluate industrial protocols on equipment under test. Those tests include communication stress, protocol robustness and other specific vulnerabilities tests.

The acquisition of an Achilles certification by a manufacturer ensures that their products and internal development processes meet industry benchmarks for security.

There are two types of Achilles certificates.

- Achilles Communications Certification: This certificate assures that manufacturers comply with the industry benchmark for the deployment of robust industrial control devices.
- Achilles Practices Certification: This certificate assures that manufacturers have followed best security management practices during their system development lifecycle. The development lifecycle comprises implementation, maintenance and decommissioning. The Achilles Practices Certification is in the process of being adopted by IEC 62443-2-4.

Completing this section, focused on the smart grid domain, we conclude that currently the standard IEC 62443 stands out among the various schemes, with ISA Secure and Achilles (both based on this standard) being the two main certificates demanded by manufacturers.

## 5.2 Safety-critical railway systems

Failure of a safety critical system can result in harm to humans and the environment. To gain confidence that the resulting risk is acceptably low, such systems have to be evaluated and certified according to applicable industry standards. To adjust the degree of scrutiny of such evaluations, these provide means to classify systems in levels of criticality with respect to the potential damage they could cause when failing. For the railway domain, the EN 50126 standard [24] introduces “Safety Integrity Levels” (SIL) that may be used as classification scheme. The SIL (SIL0 – SIL4) define different processes and methods to be obeyed during a system’s lifetime to keep its probability of failure acceptably low.

Confidence in a system, to the required degree, is gained by the so-called safety case – a structured argument, together with evidence, that compellingly provides the case for the safety of the system assuming its application in the target environment.

Within the railway sector there are numerous applications for safety-critical systems. A common element in modern environments are computer systems that perform such tasks. Adequate solutions may be realized with a flexible layered architecture consisting of an exchangeable hardware layer, the operating system and a safety middleware. The problem specific safety-critical applications run on top of this structure. An example for such a software platform is the TAS platform by Thales Rail Signalling Solutions.

To evaluate the safety of railway systems the methods described above are applied to the whole system in order to certify it. Should parts of it change, substantial effort is necessary for the re-certification, as the corresponding certification process has to be repeated for the whole system to demonstrate that safety is still guaranteed.

Regarding compositional evaluation, the state of the art in the safety domain is to define a hierarchical system architecture based on generic equipment/components, such as the TAS Platform, and subsystems. The safety case aligns well with this hierarchical approach: Each component has its own safety case which combine to form the overall safety case of the composed system. The structure of the overall safety case for a generic system is depicted in Figure 7.

The top most safety case can be seen as a generic application safety case as defined in EN 50129, whereas, for example, the concrete installation of an operation management centre is classified as specific application. In general, different kinds of safety cases are used (as supplement of the classification according to EN 50129):

- Hardware safety case

- Subsystem safety case (including hardware and software)
- Method safety case: This may be used for different purposes: Firstly, to show compliance of a concrete or principle implementation of a communication path according to EN 50159. Secondly, to provide the basis for a special methodology to achieve specific (safety) properties of an architecture.

To ensure that each component/element of an installed system can operate as specified, so-called Safety Application Conditions (SACs) are defined. As an example, such SACs may specify that the equipment is operated in the right temperature range. When such elements are used by another product or application it must be shown that the interfaces between the components and subsystems are well defined and used in compliance with their interface specification. It is essential to show the correct treatment of all SACs.

SACs can be fulfilled by a subsystem by implementing or by providing an environment, which complies with the SAC or by propagation of the SAC to the higher level system component which is using the subsystem. The evidence of the SAC will be evidenced by this higher level system.

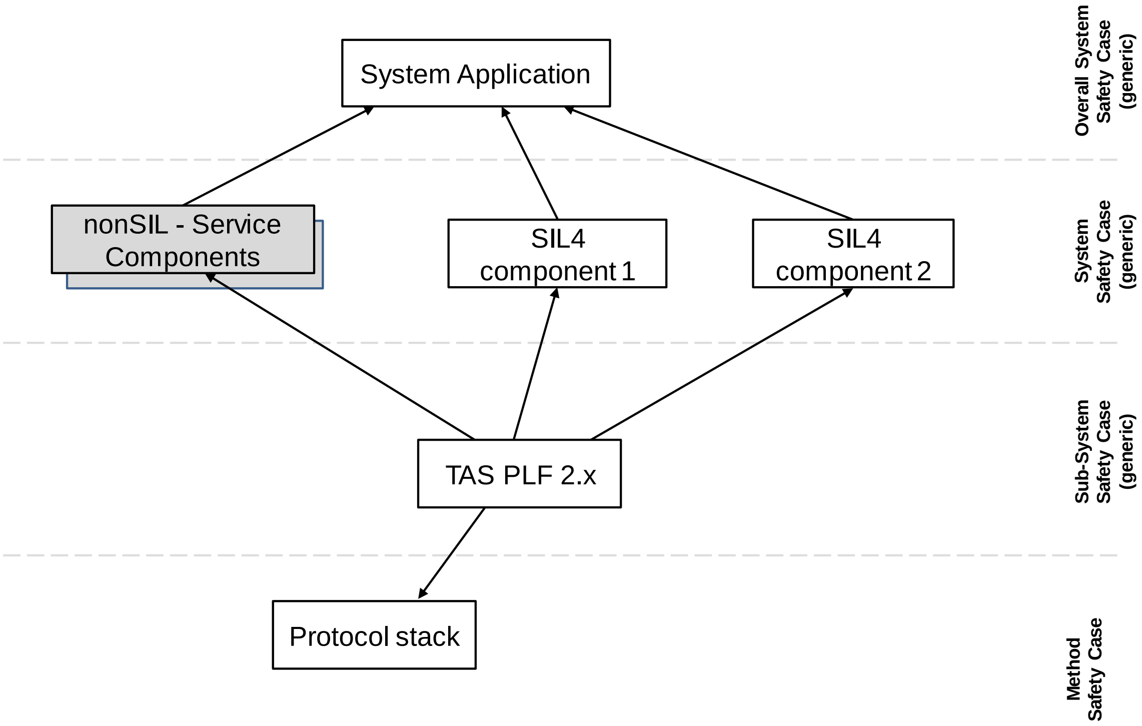
System Application
nonSIL - Service Components
SIL4 component 1
SIL4 component 2
TAS PLF 2.x
Protocol stack
Overall System Safety Case (generic)
System Safety Case (generic)
Sub-System Safety Case (generic)
Method Safety Case

Figure 7: Hierarchical structure of the safety cases [Image: THALES].

We want to reuse this safety compositional approach as basis for our security approach. With a method similar to the SACs we want to ensure that there are clean interfaces and responsibilities also with respect to security between our components, subsystems and the generic and specific applications. Note that we don’t want to mix safety and security topics, but leverage the reduced overhead of the similar approaches.

## 5.3 Application of IEC 62443 to subway systems

Urban rail transport and, in particular the subway is a highly critical infrastructure element, the malfunction of which can paralyze the operation of entire agglomeration areas. Due to the large number of passengers, it is necessary to ensure the continuous supervision of the operation of trains, track sections, stations and, in particular, of the passenger flow [25]. The disruption of the steadiness of these flows can endanger the operation of the whole transport system. It is, therefore, necessary to choose appliances and procedures such that they ensure not only the availability of the communication infrastructure, but also its security and, hence, the functional safety of the entire system [25] . While railway safety and urban guided transportation system functionalities are regulated by European directives and standards (e.g. EN 50126 and EN 62290 respectively), security issues are not fully covered. Consequently, it is necessary to identify provisions in IT- or other industrial standards on systems security.

For example, in the Czech Republic, the subway system is subordinate to the national act No. 181/2014 on cyber security. According to this law, an operator has two possible ways to fulfil its requirements. The first of option is to follow the enclosed regulations. Those include special requirements intended for control systems. They demand to ensure:

- Restriction of physical access to networks and equipment of industrial control systems,
- Restriction of interconnections and remote access to networks of industrial control systems,
- Protection of individual technological assets of industrial control systems against attacks exploiting known vulnerabilities,
- Restoration of the operation of industrial control systems after cyber security incidents.

These requirements are very generic, but at least they define the framework and delineate the issues that need to be addressed. In order to do so, however, further methodologies (standards) need to be identified to fill the gaps.

The second option permitted by the national act No. 181/2014 is to fully implement an Information Security Management System (ISMS) according to the ISO/IEC 27000 family of standards (especially ISO/IEC 27001). The drawback of this choice is its lack of requirements and guidance on the implementation for industrial automation and control systems.

With the implementation of the new Directive on security of network and information systems (NIS) [26] the situation in other EU countries can be expected to become very similar to what may be witnessed in the Czech Republic. That is, there will be a need for sector specific standards or laws concerning the cyber security of railway/subway systems. Accordingly, sources must be identified that are reasonably close to what is needed. These would be standards of the ISA/IEC 62443 (determined for the implementation of secure IACSes) and the ISO/IEC 15408 series (CC). From those sources the best suited provisions should be selected, taking into consideration the specific needs of the railway/subway operators and the used control systems.

In the first work package of the certMILS project, EZÚ, in cooperation with UCO, gathered the current interpretations of IEC 62443 and their applicability to this domain. EZÚ analysed current interpretations of this standard series, more accurately those of

- IEC 62443-1-1(Concepts and models),
- IEC 62443-2-1(Requirements for an IACS security management system),
- IEC 62443-2-3(Patch management in the IACS environment),
- IEC 62443-2-4(Requirement for IACS solution suppliers),
- IEC 62443-3-1(Security technologies for IACS),
- IEC 62443-3-3(System security requirements and security levels),
- IEC 62443-4-1 draft (Product development requirements),
- IEC 62443-4-2 draft (Technical security requirements for IACS components).

While the consideration of all above parts is important, the parts most relevant for the target application appear to be IEC 62443-3-3, IEC 62443-4-1 and IEC 62443-4-2, the last two being in draft state.

Other standards that are linked to this matter are

- IEC 62290-1 (Urban guided transport management and command/control systems - Part 1: System principles and fundamental concepts),
- IEC 62290-2 (Urban guided transport management and command/control systems - Part 2: Functional requirements specification),
- IEC 62290-3 draft (Urban guided transport management and command/control systems - Part 3: System requirements specifications) ,
- IEC 62278 (Specification and demonstration of reliability, availability, maintainability and safety, RAMS).
- IEC TR 62278-4 (Specification and demonstration of reliability, availability, maintainability and safety, RAMS - Part 4: RAM risk and RAM life cycle aspects),

- IEC TR 62267-2 (Automated urban guided transport, AUGT - Safety requirements - Part 2: Hazard analysis at top system level),
- IEC 62279 (Communication, signalling and processing systems - Software for railway control and protection systems).

A prevailing issue with the application of the individual parts of IEC 62443 is, at least in the Czech Republic, that the appliances currently employed in subway transportation are neither designed nor operated as intended by that standard series. The considered transportation system in the Czech Republic has been in operation since 1974. Although it is continuously upgraded and equipped with the most modern technologies, many important parts were constructed according to the guidelines in place at the time. The resulting historically grown system is a very complex one. For economic reasons the existing structures cannot simply be abandoned.

Therefore, it is not only unfeasible, but also not effective, to perform an an exhaustive analysis of the whole system within the certMILS project. However, it would be very helpful to perform an analysis focused on specific key domains which are particularly vulnerable within the complex system. A certified MILS platform will be the most beneficial reply to many questions that it poses.

One possible approach to this matter would be to strictly stick to the requirements of IEC 62443. An important aspect would be the general definition of the whole system according to IEC 62443-3-3 with focus on selected areas. Because the requirements of this standard could be addressed in a number of different ways, devices based on a MILS platform would create room for the implementation of services according to local security policies reusing available system services.

The most relevant part of IEC 62443 for the present use case is IEC 62443-4-2 which describes specific requirements for IACS components. The second most important part is IEC 62443-4-1 which focuses on the product development (its whole life cycle) within an organization.

Following this path, the step subsequent to the general definition of the system and its interfaces, as outlined by IEC 62443-3-3, would be an assessment according to IEC 62443-4-1. Thereafter, follows the assessment of the requirements of IEC 62443-4-2. This last step would be limited to carefully selected parts of the standard as it may proof impossible to perform the assessment against the whole standard.

The second possible approach would be wider in scope. It is similar to the one discussed above. The difference is a that other standards that are currently used (respected) would be taken into account. Those standards can form the basis on which the methodology would be built. It allows to bridge between the present situation and the state being targeted. The strongest candidate for this basis is the IEC 62290 series.

As a side note, the Czech Republic is currently a “Certificate Consuming Member” of the CC.

## 5.4 Compositional use cases for separation kernel

In this section, we provide some low-level compositional use cases, from a separation kernel manufacturer point of view.

*Common separation kernel terminology*: We adopt the terminology from [27].

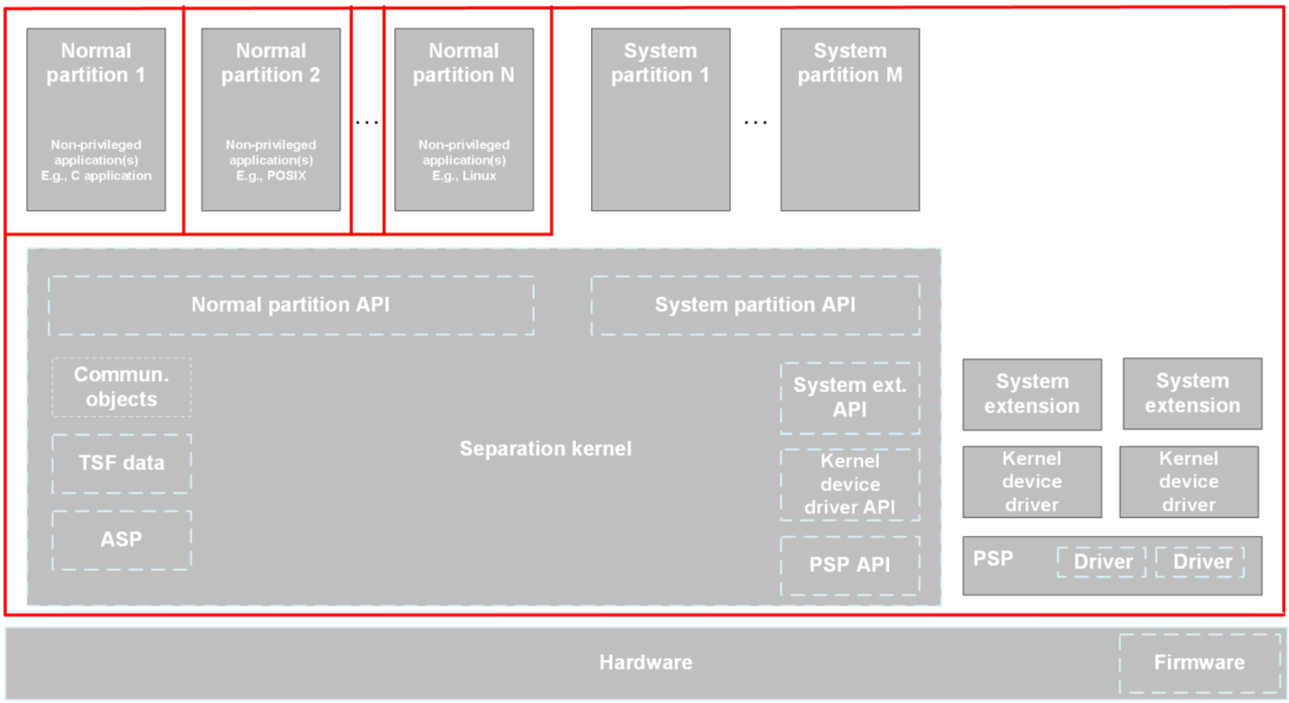

Figure 8: Generic system [Image: SYSGO].

Figure 8 shows a generic base system, similar to the one shown in the EURO-MILS PP [28]. Red lines delineate security domains established by technical means. That is, if the hardware is properly configured, an application can be confined to a normal partition. There are also other components, on the right hand of the figure, which, unless verified by the integrator, from a technical point of view, would have privileges to bypass partitioning (these will be explained below where needed).

For the life-cycle, note that a separation kernel is installed by an *integrator* to build a MILS system, before it is deployed in the field. In the field, the separation kernel runs the *applications* that have been put into *partitions* by the integrator.

### 5.4.1 Mixed criticality system

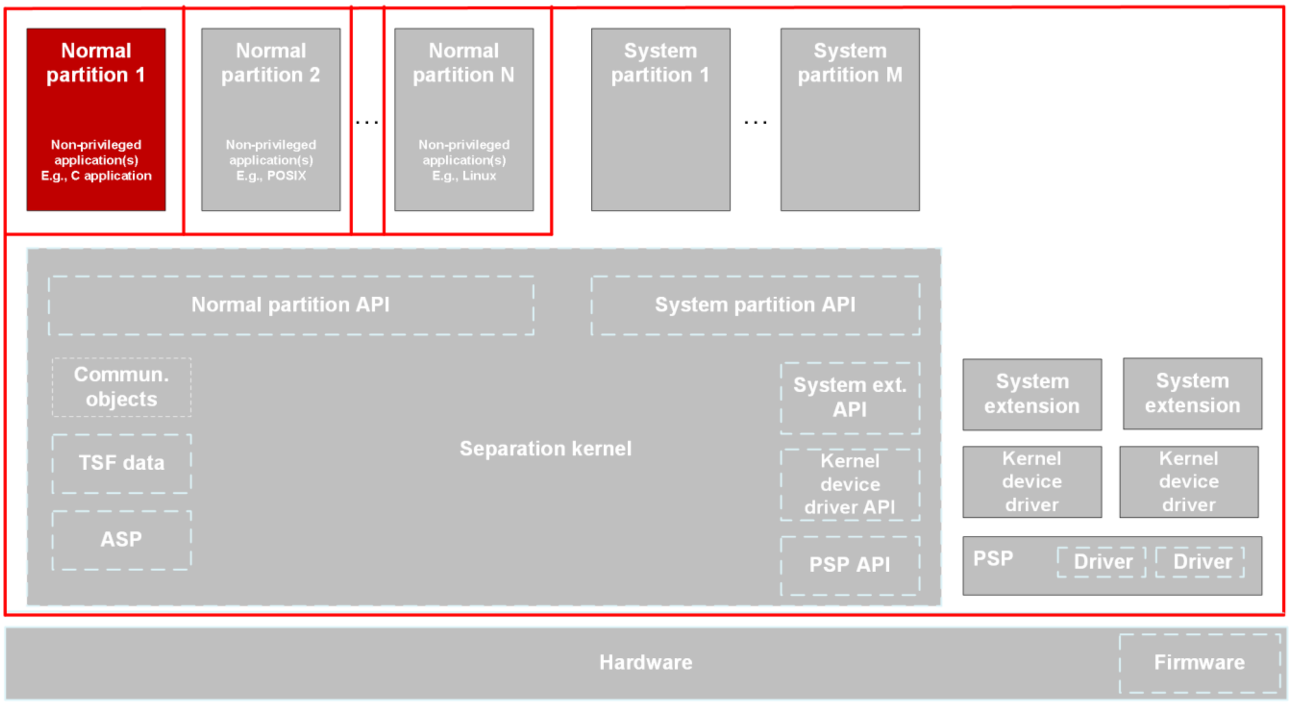

Figure 9: Mixed criticality system [Image: SYSGO]:

A typical application of separation kernels is their use as a base for access control, resource management and possibly information flow control [29] in a mixed criticality system [30]. In the example of Figure 9, normal partition 1 would be high-criticality, and it would be protected from low-criticality partition 2 to partition N by technical means provided by the separation kernel. This allows to spend less evaluation effort on partitions 2 to N. As a normal partition is built on top of the separation kernel, this kind of composition could be seen as layering composition (compare Section 3.1). When it comes to the relation between different partitions that are layered on top of the separation kernel, this kind of composition could be seen as networking composition (compare Section 3.2).

In a mixed-criticality system, applications can implement additional high-level security functionality for the system, e.g. security-critical applications or some customized auditing or watchdog functionality.

### 5.4.2 PSP (Platform support package)

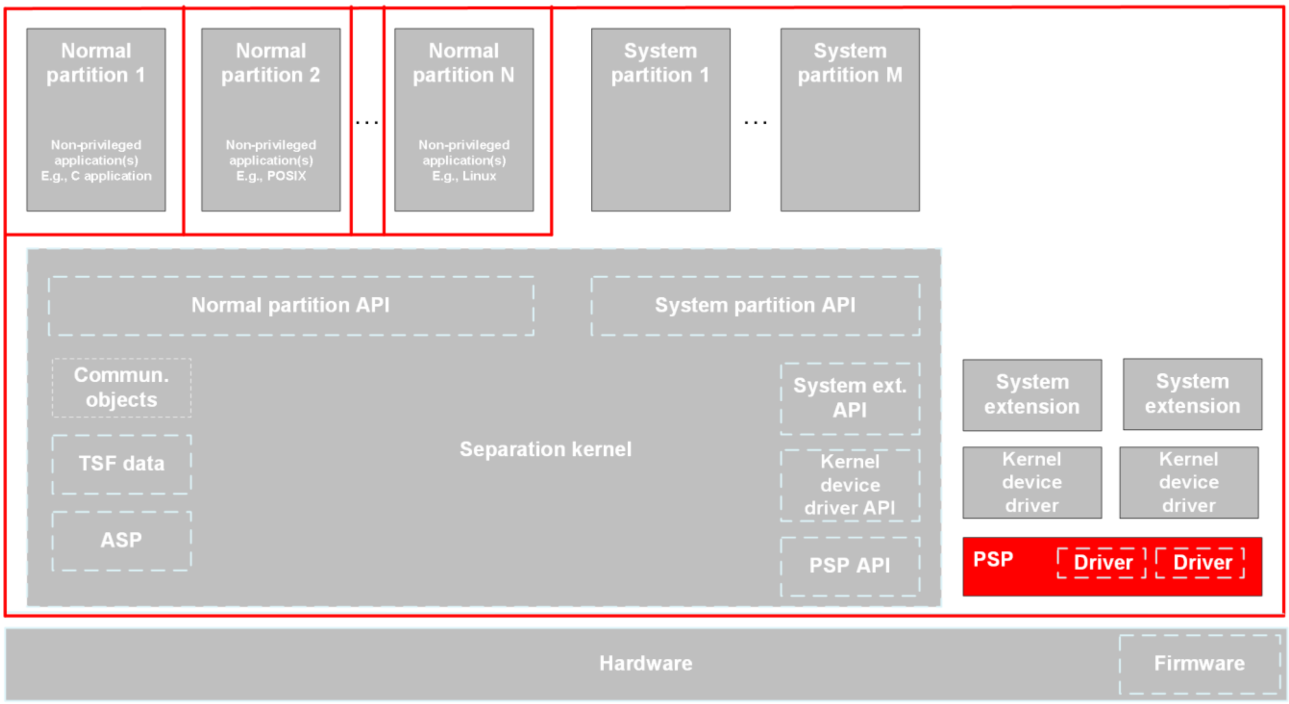

Figure 10: PSP [Image: SYSGO].

For desktop computers, at initialization, often there is a BIOS, UEFI, or similar, creating a comparatively uniform and well-described environment that a desktop operating system will run on. Thus, when installing a desktop operating system, it often suffices to select a CPU architecture. Embedded systems often lack this infrastructure (also for performance reasons) and some board-specific initialization parameters, for instance an initial mapping of memory and interrupts, have to be hard-coded. This is usually achieved by a platform support package (PSP).

That is, the PSP contains a set of drivers for specific hardware components and is supplied and approved by the integrator. A PSP uses the separation kernel's PSP API. In operational use, the product based on the separation kernel always contains exactly one PSP. When the hardware reboots, the PSP does some initialization. At run-time, the PSP may provide some interrupt service routines. The PSP is in the same address space and in the same technical security domain as the separation kernel.

However, a PSP usually will not bring any new security functionality. In terms of the Common Criteria, a PSP is a non-bypassing piece of software.

Formally, a PSP is in the same address space as the separation kernel, the component may influence the other via channels that differ from the intended ones, this type of composition could be seen as component composition, as defined in Section 3.3. On the other hand, the API is well-defined, and the component is small and can be checked by the integrator, so it also could be seen as a layering type composition discussed in Section 3.1. certMILS also considers to investigate whether this composition could/should be represented by use of a modular ST / modular PP.

### 5.4.3 Partitioned network card

For networks such as Ethernet, many network cards allow to join the network with multiple virtual media access control (MAC) addresses. In many designs, each virtual MAC address has its own network queues in hardware [31]. Such hardware designs can be virtualized by drivers which translate the interfaces of the network queues and control commands for the network cards to suitable user space interfaces. As the interfaces for queues, and card control can be quite complex, writing such a driver can be a complex and thus error-prone undertaking. Thus, in systems where network connectivity has lower criticality than other functionality, it is possible to contain a virtualizing network driver, including its complex code-base, in a normal (i.e. non-privileged) partition (as a special case of mixed-criticality system, see Figure 9). For example, for cars, the ability to locally apply the brakes is more critical than the ability to connect to the Internet.

As a normal partition is built on top of the separation kernel, this kind of composition could be seen as layering composition (compare Section 3.1). When it comes to the relation between the partition hosting the network card driver and other different partitions that are layered on top of the separation kernel, this kind of composition could be seen as networking composition (Section 3.2).

### 5.4.4 Kernel-level driver for some communication protocol

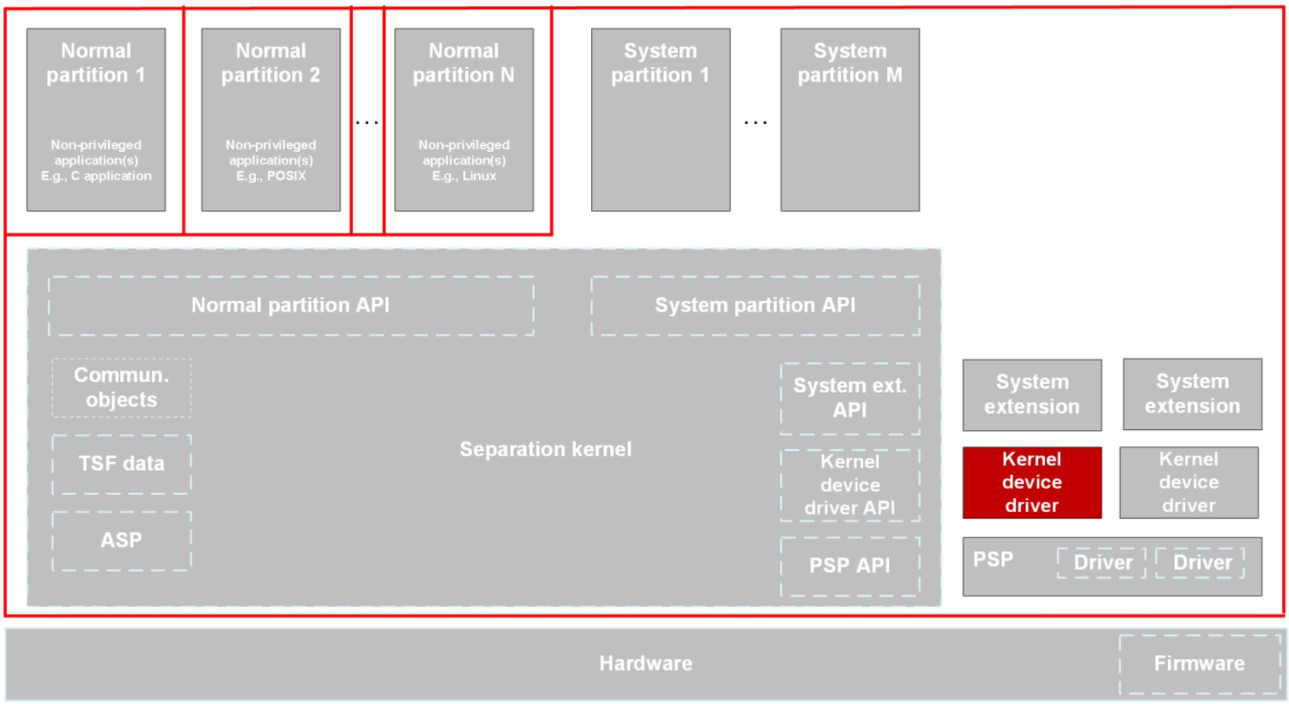

Figure 11: Kernel-device driver [Image: SYSGO].

The certMILS separation kernel provides several communication mechanisms, such as ports, files, shared memory, events, and IPC. To build a MILS system, an integrator installs different applications on the separation kernel. When the MILS system is in operational use, applications request separation kernel services at run-time. Thus, for each communication mechanism there can be an integration time interface and a run-time interface.

The separation kernel is small, thus it natively supports only a limited sets of communication protocols. If desired by the user, it is however possible to add additional protocols. In the following we assume that some specific communication protocol is to be supported. The easier and safer option is to use the already existing communication primitives and to write the driver in the application address space. However, in exceptional cases it might be desirable to have the protocol within the address space of the separation kernel itself by realizing an appropriate kernel-level driver (see

Figure 11). If the communication protocol is small, then the complexity of the driver would largely be given by configuration interfaces that are intended for the user. Such interfaces could be based

- on the integration time API, which is exposed to the integrator,
- and/or on run-time services, which are exposed to applications at run-time.

Such a use case can be security-critical, so that the kernel-device driver should be certified. Therefore, drivers of this kind could represent an application for compositional certification of a possibly small component based on a separation kernel.

As a kernel device driver resides in the same address space as the separation kernel, the component may influence others via channels that differ from the intended ones, this type of composition could be seen as component composition, as defined in Section 3.3. On the other hand, the API is well-defined, and if the component is small and can be checked by the integrator, it also could be seen as layering composition, described in Section 3.1. Unlike in Section 5.4.1, where a critical application could implement new security functionality, in the present case the composition does not add extra security functions.

### *5.4.5 I/O MMU as optional hardware component*

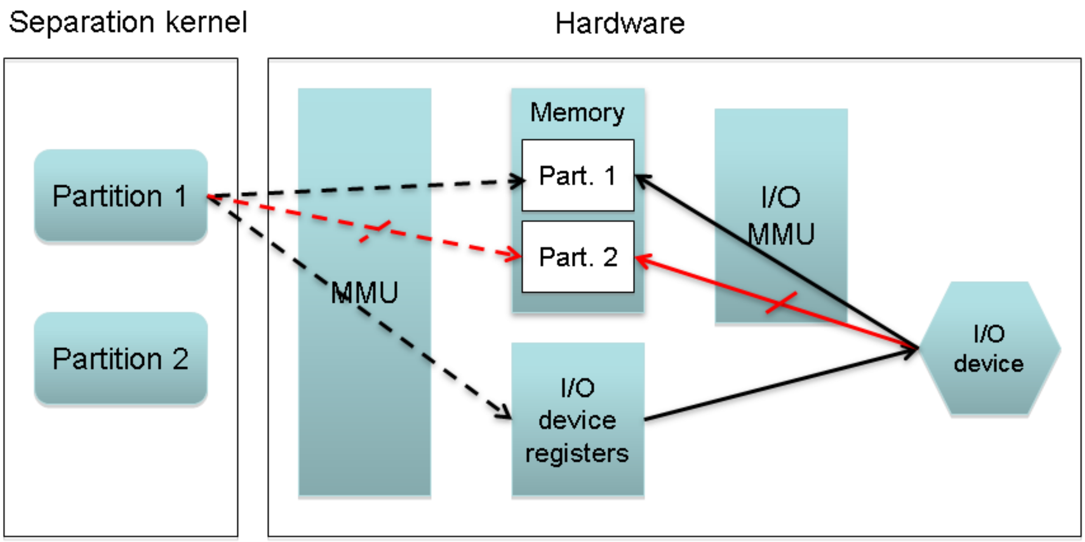

Figure 12: MMU and I/O MMU configuration example [Image: EURO-MILS].

COTS CPUs come with an MMU, which controls the access to memory. Figure 12 illustrates how the I/O MMU comes into play. The separation kernel sets up the MMU, based on a configuration provided by the integrator. In Figure 12, the access control configured for the MMU - enforced for partition 1 - is depicted by dashed arrows. The MMU has been configured so that partition 1 can access part of the memory (upper dashed black arrow to box labelled “Part. 1”). Further, partition 1 does not have to access to other parts of the memory (dashed red arrow to box labelled “Part. 2”, in the present example this memory is reserved for partition 2).

While the MMU controls access from partitions to memory-mapped I/O-registers (this is exemplified by the lower dashed black arrow in Figure 12), the I/O MMU controls access of the return path from I/O devices to the partitions’ memory segments. A typical use case for I/O devices is to provide DMA writing data back to the memory of the subject that controls the I/O device (in this case the memory of Partition 1, in the figure this is indicated by the upper solid black arrow). However, once a device uses DMA, it has to be ruled out that it writes to the wrong memory part (in this case memory assigned to partition 2, indicated by the plain red crossed arrow). DMA control is a typical I/O MMU application: it allows an I/O device only to access certain memory regions.

As I/O MMUs are more recent and less standardized than MMUs, i.e. not every system has an I/O MMU, it makes sense to split the I/O MMU functionality off in the evaluation and certification of the separation kernel.

On the software side, as there is no direct dependency between MMU and I/O MMU, it is possible to encapsulate the software for the control of the I/O MMU into a different subsystem/module than that for the control of the MMU. Whether an I/O MMU driver is more efficiently implemented as a kernel-device driver, in a PSP, or otherwise, depends on system and hardware properties.

Thus it can be beneficial to evaluate the software of the I/O MMU independently of that of the MMU. Alternatively to CAP (Section 4.1) and CPE (Section 4.2), another conceivable way to solve the problem of evaluating a system optionally together with an I/O MMU is to use a modular ST / modular PP. This topic is addressed in certMILS work packages 2 and 3.

# Chapter 6 Specific Issues for certMILS

## 6.1 The MILS concept

Numerous IT applications in the private consumer, industrial and military sector require the processing of information at different security levels. That is, they demand systems that provide access to users with diverse background and varying security clearance (mixed-criticality systems), without spoiling any of their security provisions. Multi Level Secure (MLS) systems achieve this by separating security domains on different levels.

MLS systems can consist of several connected computers or, where efficiency is an important concern, be realized by specialized operating systems that run on single machines. The downside of the latter approach is that such software quickly becomes too complex to be assessed in security evaluations. In the absence of stringent separation, all parts, be they relevant for the systems security functionality or not, must be considered in evaluations. If the demanded degree of security assurance is high this problem becomes particularly pressing, and the economic realization of such systems may become impossible.

The Multiple Independent Levels of Security (MILS) concept, perceived already in 1984 [32], addresses this issue with a divide-and-conquer approach. It separates the critical security functionality in dedicated layers with a compact code-base. This critical functionality can e.g., be realized by separation kernels enabling this principle. They take the primary task of separating applications “in space and time”. This means that they provision hardware like memory or CPU time, needed by all applications, such that they cannot get into causal contact.

Other functionality of lower security criticality may need to be provided by middleware that runs on top of the separation kernel, as its inclusion would bloat the code base of the separation kernel.

This leads to a layered structure, possibly consisting of hardware, separation kernel, middleware and application layers. Each of the layers depends on security features of other layers, hence there are interactions. For example, the separation kernel may depend on that the hardware guarantees separation of physical memory compartments, but the separation kernel may also need to configure the hardware to this end.

The separation kernel, in turn, needs to guarantee its security features to the middle-ware or applications running on top. It has to provide virtual isolated domains (called partitions in Section 5.4) and restrict possible communication between these to a well defined set. To achieve the properties expected of a MILS system the separation kernel has to provide (together with the single piece of hardware on which it runs) [33]::

- Data Separation: Applications in one partition cannot access private data of others.
- Control of Information Flow: Restriction of information exchange between partitions to authenticated and authorized transfer.
- Fault Isolation: Prevention of damage from propagation of errors to other partitions.
  Temporal Separation: Sharing of resources like CPU by applications with time slicing (“periods processing”).

As it takes the burden of implementing most security critical functionality, among the software layers of a MILS system, the separation kernel will be subject to particular scrutiny. The ambitious properties required of such specialized operating systems have become known as NEAT principles: non-bypassable (its security functions cannot be circumvented), evaluatable (its code-base is small enough to allow evaluation of the correctness of the security functions), always invoked (at any time are its security checks active) and tamperproof (it prevents any unauthorized change).

While major security functions of a MILS system may be concentrated in a separation kernel, it should be clear from the description above that it will always depend on functionality provided by other layers. In an independent CC evaluation of a separation kernel these dependencies would be expressed in assumptions made about the operational environment of the kernel. However, for security critical applications it may be necessary to evaluate a larger part of the MILS systems, such as the separation kernel in combination with the drivers needed to execute the kernel on a particular hardware platform (for instance PikeOS together with a corresponding PSP; compare Section 5.4.2). Eventually, without these components, a separation kernel cannot provide its security functions.

## 6.2 Compositional evaluation of MILS systems

The above considerations, concerning the layered structure of MILS systems, may suggest to approach the assessment of such combinations as compositional evaluation for systems of layering type (compare Section 3.1), but this approach has its limitations since the interaction between the layers of a MILS system is not strictly one-directional. This aspect of bi-directional communication in a constrained and predictive way would better be addressed as a composed system of network type (compare Section 3.2). See Section 5.4 for some possible use cases of compositional evaluation in this context.

A separation kernel allows to confine the interactions between partitions to precisely defined communication channels. Once these have become permanent, a partition can defend its integrity against others. Also the mechanisms provided by such a kernel can ensure, once and for all, that there is no illicit read accesses to partition memory. While it is almost always theoretically possible to bypass the access control policies tampering with the hardware setup, such a configuration would have to be explicitly enabled by the integrator. It appears that an integrator with moderate understanding of the hardware platform can therefore fend off this threat.

However, experience with the layering type of evaluation applied to smart cards (compare section 4.2) has shown the significance of deriving the detailed technical objectives for the lower layer (in the case of the smart cards: the smart card hardware platform and embedded firmware). Since the security objectives of, and assumptions made by, the higher (application) layer are usually not known when performing the evaluation of the lower one, tentative assumptions about the detailed technical security objectives for the application have to be made. History has shown that this fails quite often, resulting in critical security issues for the composed system. See [7] for some examples.

A problem that came up quite early in smart card evaluations is the existence of side channels that can be exploited by an attacker to extract cryptographic key material or plain text. There are many possible side channels that are relevant for smart cards, such as their power consumption, emanation, timing, and fault injection. The sources of all these lie within the hardware. Therefore, the detailed technical security objectives an application implementing cryptographic algorithms would have to impose on the hardware to address the issue lead to requirements that are impossible to satisfy with present day technology.

This is a typical example where the interdependencies between two layers become too complex to manage when the decomposition is done such that the implementation of the cryptographic algorithms is part of the application layer. It is for this reason that many modern smart cards are delivered with the cryptographic algorithms implemented as part of the embedded firmware of the platform. This implementation can and does take the physical characteristics of the hardware into account and incorporates countermeasures against side channel attacks, built on the detailed knowledge of hardware characteristics. With respect to the evaluation, this reduces the amount of detailed technical security objectives applications need to define for the lower layer - provided the application does not implement functions for which side channels of the underlying hardware/firmware layer can result in critical vulnerabilities.

As pointed out in section 5.4.5, reducing the interdependencies between components with respect to the security properties / security objectives is essential to constrain the complexity of the compositional evaluation and to simultaneously achieve a high level of assurance. Therefore, in theory, the overall security objectives of a system should be taken into account when defining its components. In practice this is not always possible since the components to be used oftentimes already exist when building a system. Consequently, most components have to be evaluated anticipating the detailed security objectives.

Looking into the example of MILS separation kernels, we identify the following set of components that may be subject to component evaluation:

- The kernel itself (in the following referred to as simply the “kernel”),
- the platform support package (PSP),
- the kernel device drivers,
- the system/kernel extensions,
- the privileged partitions,
- normal partitions,
- parts of the hardware like an I/O MMU.

Within a MILS architecture, which may have several of these components, the kernel plays a central role (compare section 6.1). It is also the one component that may exist before others are developed. Henceforth, an evaluation of the kernel has to anticipate what the type of security properties normal partitions expect it to provide and the detailed technical security objectives for the PSP and the kernel device drivers have to be defined. In addition, the PSP developer may need to define limitations for system extensions and kernel device drivers.

In order to satisfy specific security objectives, like fault tolerance or non-interference, the kernel needs to implement its functions such that those properties can be satisfied. This also results - as stated above - in specific technical security objectives and constraints for other components. It is obvious that those are dependent on the overall security objective and there may well be cases where two different objectives for the whole system can result in contradicting detailed technical security objectives for the components. Fault tolerance and non-interference are examples of such high level objectives where the system may not be able to satisfy both of them at the same time completely.

While there may be different configurations of the kernel that satisfy each of them separately (resulting in specific detailed technical objectives and constraints for the PSP, device drivers, system extensions and privileged partitions), to satisfy both a compromise may be needed that implements each of the properties with some limitations to resolve contradictions. Those limitations have to be elaborated in the evaluation and described as residual risks that need to be addressed by the operational environment, unless the system operator is willing to accept them. Even when the system is supposed to satisfy just one of the security objectives mentioned above there is usually a residual risk that this objective cannot be fully satisfied. There are always technical limits to both fault tolerance and non-interference that need to be identified and described as part of the evaluation.

# Chapter 7 Possible Approaches for certMILS

### 7.1.1 Integration of partitions/applications on top of the separation kernel

Starting from the separation kernel, certMILS will work out a further strategy in deliverable D1.3. It is in the interest of certMILS that composition approaches can be deployed that make the verification effort feasible for an integrator, especially for layering applications on top of the separation kernel – it is to be feared that otherwise attempts at secure composition might be abandoned from the beginning. Of the composition types, layered composition appears to be the most light-weight one, but not all elements of a MILS architecture are characterized by a uni-directional use of functions. Therefore, some aspects may have to be modelled akin to the network type of composition. Note again that, in general, a system will be of mixed type. The smart card approach to compositional evaluation has such properties.

For the realization of the described types of composition, it is crucial that all interaction between the components is restricted to well defined channels. Possible exceptions need to be uncovered in the course of an evaluation. However, when a component of a system is developed it is usually not precisely known how other parts are going to use it. The developer of the application (i.e. the integrator) may therefore require additional information about the platform's implementation to ensure that there are no exploitable covert channels.  If such a mixed or smart card type approach is to be adopted, it, in particular, could be backed by design guidance to the integrator as part of deliverable D2.4.

### 7.1.2 Composition of a PSP or I/O MMU with the separation kernel

Since, for layered structures, the assurance level achieved for the composed system cannot plausibly exceed that of lower layers, high assurance applications require particular scrutiny when it comes to the evaluation/certification of the platform. With this ambition the well-definedness of interactions between components becomes particular pressing and the compositional evaluation approach more involved. Therefore, for the composition of, for example, a separation kernel with a PSP and/or an I/O MMU stronger composition approaches may be needed. If such methods happen to be impossible, or without benefits compared to a monolithic evaluation of the composed TOE, the approach of modular PPs may at least provide flexible means to equip integrators with customizable templates for their security targets. It is left for further work in certMILS to establish whether modular PPs can aid the development of applications by integrators . Again, this approach could be backed by deliverable D2.4.

### 7.1.3 Zoned systems: IEC 62443

If we step up looking at larger systems, IEC 62443 tackles these by zoning of networked components. Again, it appears that this kind of system might be modelled as network type composition so that, when viewed top-down, it looks as if design guidance as anticipated for certMILS deliverable D2.4 will be reusable for IEC 62443 and other high-level standards. The certMILS project provides the opportunity to test this hypothesis by the iteration through the concrete demonstrators.

### 7.1.4 Verification methods for composition

In addition to the possible need for transfer of design guidance, a further topic is the efficient verification/testing of composed systems to ensure that security policies will not be bypassable by the added components. In the example of the composition of a separation kernel with a PSP this could cover the development of a PSP test suite, verifying that the separation kernel policies are not modified/bypassed, or the fuzzing of PSP interfaces.

# Chapter 8 Summary and Conclusion

The present report concerned itself with the documentation of the current regulatory situation of various CPS and the applicability of compositional evaluation strategies to MILS systems. As discussed, that situation is diverse. In both cases there is a number of different approaches but definite solutions are scarce goods in all considered domains.

The focus of this account was the regulative baseline with respect to compositional evaluation to explore whether it has benefits for the certification of safety-critical applications. The bottom line here is that there is a framework which is successful in the high-assurance evaluation of smart cards, but also has its shortcomings. Unintended interactions between components might be overlooked in evaluations if the developer of the composed system has only limited design information for the components.

Another approach, which is part of the CC, is more generic but allows to reach only relatively low assurance levels. It has not found widespread application. Hence, there is no silver-bullet among the existing strategies. A novel framework which achieves high assurance but circumvents the pitfalls would likely be applicable to specific types of systems (such as the smart card CPE approach) and require increased information transfer between developers of components and composed systems.

By the example of three different certMILS pilots, the present document also provided an overview of the intricate situation that large-scale CPS face when it comes to safety and security concerns. There is a variety of standards, some of which are under active development, but the state of the art in technology in the different domains seems hardly accounted for.

The previous analysis shows that it is necessary to investigate the components of MILS systems for applications like the mentioned pilot projects, with respect to their independent security properties and how these combine. That is, certMILS will, in a bottom up approach, tackle those security properties that can be composed, to the extent possible. certMILS shall also investigate how claims on compositional assurance can be backed by an adequate compositional assurance methodology.

# Chapter 9 List of Abbreviations

| Abbreviation | Translation |
|---|---|
| AUGT | Automated Urban Guided Transport |
| BCU | Bay Controller Units |
| BIOS | Basic Input/Output System |
| CASCO | Committee on Conformity Assessment (ISO) |
| CC | Common Criteria |
| CCDB | Common Criteria Development Board |
| CCRA | Common Criteria Recognition Arrangement |
| COTS | Commercial Off-The-Shelf |
| CPS | Cyber Physical System |
| CPU | Central Processing Unit |
| DMA | Direct Memory Access |
| EAL | Evaluation Assurance Level |
| EDSA | Embedded Device Security Assurance |
| ETR | Evaluation Technical Report |
| FR | Foundational Requirement |
| I/O | Input-Output |
| IACS | Industrial Automation and Control Systems |
| I&A | Identification and Authentication |
| IEEE | Institute of Electrical and Electronics Engineers |
| IEC | International Electrotechnical Commission |
| IED | Intelligent Electronic Devices |
| IO | Input-Output |
| IP | Internet Protocol |
| IPC | Inter-process Communication |
| ISA | Instrumentation Systems and Automation Society |

| | |
|---|---|
| ISCI | ISA Security Compliance Institute |
| ISMS | Information Security Management System |
| ISO | International Organization for Standardization |
| IT | Information Technology |
| LDAP | Lightweight Directory Access Protocol |
| MAC | Media Access Control |
| MILS | Multiple Independent Levels of Security |
| MLS | Multi Level Security |
| MMU | Memory Management Unit |
| PP | Protection Profile |
| PSP | Platform Support Package |
| SAC | Safety Application Condition |
| SAL | Security Assurance Level |
| SDLA | Security Development Lifecycle Assurance |
| SFR | Security Functional Requirement |
| SIL | Safety Integrity Levels |
| SSA | System Security Assurance |
| ST | Security Target |
| TOE | Target of Evaluation |
| TSF | TOE Security Function |
| UEFI | Unified Extensible Firmware Interface |

# Chapter 10 Bibliography

[1] K. Müller, “Trustworthy MILS: CC Composite Evaluation Approach,” 2015. [Online]. Available: http://dx.doi.org/10.5281/zenodo.47300.

[2] CCMB, “Common Criteria for Information Technology Security Evaluation v3.1, Part 1: Introduction and general model,” 2017. [Online]. Available: https://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf.

[3] CCMB, “Common Criteria for Information Technology Security Evaluation v3.1, Part 3: Security assurance requirements,” 2017. [Online]. Available: https://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.pdf.

[4] Joint Interpretation Library, “Composite product evaluation for Smart Cards and similar devices,” 2015. [Online]. Available: https://www.sogis.org.

[5] D. Craigen and M. Saaltink, “Review of the Composability Problem for System Evaluation,” *Defence R&D,* 2004.

[6] CCMB, “Common Methodology for Information Technology Security Evaluation v3.1,” 2017. [Online]. Available: https://www.commoncriteriaportal.org/files/ccfiles/CEMV3.1R5.pdf.

[7] P. Karger and H. Kurth, “Increased Information Flow Needs for High-Assurance Composite Evaluations,” *Second IEEE International Information Assurance Workshop,* pp. 129-140, 2004.

[8] I. Furgel, V. Saftig, T. Wagner, K. Müller, R. Schwarz and A. S. F. Blomberg, “Non-Interfering Composed Evaluation,” 2016. [Online]. Available: http://dx.doi.org/10.5281/zenodo.47979.

[9] International Electrotechnical Commission, Technical Committee 65: Industrial-process measurement and control, “IEC 62443: Security for industrial automation and control systems,” 2008. [Online]. Available: http://www.iec.ch/.

[10] ISA Security Compliance Institute, “ISASecure - IEC 62443-4-1 - SDLA Certification,” 2017. [Online]. Available: http://www.isasecure.org/en-US/Certification/IEC-62443-SDLA-Certification.

[11] ISA Security Compliance Institute, “ISASecure - IEC 62443-4-2 - EDSA Certification,” 2017. [Online]. Available: http://www.isasecure.org/en-US/Certification/IEC-62443-EDSA-Certification.

[12] “Minimizing cyber security risks with a trusted and flexible partner for future requirements,” 2017. [Online]. Available: http://w3.siemens.com/smartgrid/global/en/products-systems-solutions/cyber-security/Pages/wib-certification.aspx.

[13] ISA Security Compliance Institute, “SDLA-312 Security Development Lifecycle Assessment,” 2017.

[14] P. Theron, “Introduction to the European IACS components Cybersecurity Certification Framework (ICCF),” 2016. [Online]. Available: https://doi.org/10.2760/717579.

[15] “European Technology Platform Smart Grid,” 2017. [Online]. Available: http://www.smartgrids.eu/.

[16] “Analysis of the Cyber Attack on the Ukrainian Power Grid,” 2016. [Online]. Available: https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf.

[17] “ISO/IEC 27001:2013, Information technology - Security techniques - Information security management systems - Requirements,” 2017. [Online]. Available: https://www.iso.org/standard/54534.html.

[18] “ISO/IEC TR 27019:2013, Information technology - Security techniques - Information security management guidelines based on ISO/IEC 27002 for process control systems specific to the energy utility industry,” 2017. [Online]. Available: https://www.iso.org/standard/43759.html.

[19] “ISO/IEC 15408-1:2009, Information technology - Security techniques - Evaluation criteria for IT security - Part 1: Introduction and general model,” 2017. [Online]. Available: https://www.iso.org/standard/50341.html.

[20] IEEE Power and Energy Society, “1686-2013 - IEEE Standard for Intelligent Electronic Devices Cyber Security Capabilities,” 2014.

[21] “ISA Security Compliance Institute,” 2017. [Online]. Available: www.isasecure.org.

[22] “International Oganization for Standardization,” 2017. [Online]. Available: www.iso.org.

[23] “Common Criteria Portal, Schemes,” 2017. [Online]. Available: https://www.commoncriteriaportal.org/ccra/schemes/.

[24] *EN 50129: Railway applications. Communication, signalling and processing systems. Safety related electronic systems for signalling,* 2003.

[25] P. Novobilsky, T. Kertis, D. Prochazkova and J. Proch, “Cyber security of metropolitan railway communication infrastructure,” in *FVTM UJEP*, Usti nad Labem, 2016.

[26] “DIRECTIVE (EU) 2016/1148 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL,” 2016. [Online]. Available: http://data.europa.eu/eli/dir/2016/1148/oj.

[27] S. Tverdyshev, H. Blasum, B. Langenstein, J. Maebe, B. De Sutter, B. Leconte, B. Triquet, K. Müller, M. Paulitsch, A. Söding-Freiherr von Blomberg and A. Tillequin, “MILS Architecture,” 2013. [Online]. Available: http://dx.doi.org/10.5281/zenodo.45164.

[28] I. Furgel and V. Saftig, “D12.3 Common Criteria Protection Profile, "Multiple Independent Levels of Security: Operating System" (MILS PP: Operating System),” 2016. [Online]. Available: https://doi.org/10.5281/zenodo.51582.

[29] J. Rushby, “Separation and Integration in MILS (The MILS Constitution),” no. SRI-CSL-08-XX, 2008.

[30] A. Burns and R. I. Davis, “Mixed Criticality Systems - A Review,” 2017. [Online]. Available: http://www-users.cs.york.ac.uk/burns/review.pdf.

[31] J. Renato Santos, Y. Turner and J. Mudigonda, “Taming Heterogeneous NIC Capabilities for I/O Virtualization,” 2008. [Online]. Available: https://www.usenix.org/legacy/events/wiov08/tech/full_papers/santos/santos_html/index.html.

[32] J. Rushby, “A Trusted Computing Base for Embedded Systems,” *Proceedings of the 7th D/NBS Computer Security Conference,* 1984.

[33] C. Taylor, M. W. Vanfleet, J. A. Luke, W. R. Beckwith, C. Taylor, B. Calloni and G. Unchenick, “Mils: architecture for high-assurance embedded computing,” *CrossTalk: The Journal of Defense Software Engineering,* 2005.

[34] CCMB, “Common Criteria for Information Technology Security Evaluation v3.1, Part 2: Security functional requirements,” 2017. [Online]. Available: https://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R5.pdf.