Security and Safety Modelling

Artemis JU Grant Agreement no.: 295354

Integrated Design and Evaluation Methodology

30 April 2014

Dissemination level: PU

Contributing partners

ADEL, CTU, EADS, GM, IFAG, IKV, INT, SAG, SYSGO, UC
# TABLE OF CONTENTS

1. Introduction ................................................................................................................. 10
2. Concepts and Principles ............................................................................................... 11
   2.1 Terminologies and taxonomies .................................................................................. 11
       2.1.1 Security terminology ....................................................................................... 12
       2.1.2 Safety terminology ....................................................................................... 12
       2.1.3 Discussion ..................................................................................................... 13
       2.1.4 Impact on the SESAMO generic process .......................................................... 15
   2.2 Levels and classification ............................................................................................. 15
       2.2.1 Safety integrity levels ....................................................................................... 15
       2.2.2 Security levels ............................................................................................... 16
       2.2.3 Evaluation assurance levels ............................................................................. 16
       2.2.4 Discussion ..................................................................................................... 17
       2.2.5 Impact on the SESAMO generic process .......................................................... 17
   2.3 Risks and levels: Techniques, measures, and controls ............................................... 18
       2.3.1 Impact on the SESAMO generic process .......................................................... 19
   2.4 Tolerability of risk and ALARP .................................................................................. 19
       2.4.1 Impact on the SESAMO generic process .......................................................... 20
   2.5 Safety cases ............................................................................................................. 20
       2.5.1 Impact on the SESAMO generic process .......................................................... 23
3. SESAMO Generic Process ............................................................................................. 24
   3.1 Introduction ............................................................................................................. 24
   3.2 Safety Lifecycle (IEC 61508) ................................................................................... 26
       3.2.1 Safety Lifecycle Overview .............................................................................. 26
       3.2.2 Concept Phase ............................................................................................... 27
       3.2.3 Overall Scope Definition ............................................................................... 28
       3.2.4 Hazard and Risk Analysis .............................................................................. 28
       3.2.5 Overall Safety Requirements ......................................................................... 29
       3.2.6 Overall Safety Requirements Allocation ......................................................... 29
       3.2.7 Overall Safety Requirements Specification ..................................................... 30
       3.2.8 Safety-Related Systems Realization ................................................................. 30
   3.3 Role Definitions ........................................................................................................ 32
   3.4 Formalized Generic Process Definition .................................................................... 37
       3.4.1 Concept of Operations .................................................................................... 38
       3.4.2 System Design ............................................................................................... 44
       3.4.3 Development ................................................................................................. 51
       3.4.4 Safety and Security Validation ....................................................................... 54
   3.5 Relationship to SESAMO Methods and Building Blocks .......................................... 55
## Model-based Perspective

4.1 Contributions to the SESAMO model-based perspective from CHESS

4.1.1 SafeCer extensions to CHESS for safety arguments

4.2 Modelling formalisms

4.2.1 UML-based formalisms

4.2.2 Other modelling formalisms

## Assurance Perspective

5.1 Risk-Management

5.2 Assurance and evidence

5.3 Decision support

5.4 Security-Informed safety cases

5.5 Security-informed risk assessment

5.6 Security-informed safety case methodology

5.7 Layered approach

## Industrial Application

6.1 Automotive Adaptation

6.1.1 Lifecycle of the automotive safety standard ISO 26262

6.1.2 Approach for a safety-security combined process definition

6.2 Avionics Adaptation

6.2.1 Safety life cycle

6.2.2 Security life cycle

6.2.3 Integration between security and safety risk assessment

6.2.4 Integration of security development activities in the development process

6.3 Railway Adaptation

6.3.1 The lifecycle according to EN 50126

6.3.2 Overall approach to risk management in EN 50126

6.3.3 Functional safety integrity for E/E/EP

6.3.4 An example of THR explicit risk estimation

6.3.5 The railway “building blocks and analysis methods” [72]

6.3.6 Security approaches

6.4 Industrial Process Control Adaptation

6.4.1 Scope Exploration

6.4.2 Hazard and Risk Analysis

6.4.3 Requirements Derivation

6.4.4 System Design

6.4.5 Hardware and Software Design

6.4.6 Development

## Discussion

7

## Summary and Conclusions

8
LIST OF FIGURES

Figure 1 – Model of system failure behaviour.................................................................13
Figure 2 – The ALARP principle......................................................................................20
Figure 3 – Toulmin’s formulation of a claim.................................................................22
Figure 4 – Risk management process (from ISO 27005).................................................24
Figure 5 – Risk treatment activity (from ISO 27005)....................................................25
Figure 6 – Safety Lifecycle – Overview.........................................................................26
Figure 7 – Industrial Domain Safety Lifecycle – Detailed.............................................32
Figure 8 – Generic Process Definition............................................................................38
Figure 9 – Concept of Operation....................................................................................39
Figure 10 – Interleaved activities....................................................................................45
Figure 11 – System Design.............................................................................................46
Figure 12 – Hardware and Software Design....................................................................49
Figure 13 – Development.................................................................................................52
Figure 14 – Safety and Security Validation.....................................................................54
Figure 15 – Extended V-model in the CHESS / SafeCer development process.............61
Figure 16 – Model-based approach in the avionics domain (see also Appendix A) .................63
Figure 17 – Automotive Domain Safety Lifecycle.................................................................69
Figure 18 – V-Model applied in ISO 26262 (from ISO 26262 standard)............................70
Figure 19 – Formalized V-Model for a safety-security combined process............................71
Figure 20 – Safety and security activities of the Concept Phase .........................................72
Figure 21 – Safety and security activities of the System Design Phase ...............................74
Figure 22 – Safety and security activities of the Hardware/Software Design Phase ...............75
Figure 23 – Safety and security activities of the Hardware/Software Design Phase ...............76
Figure 24 – Overview of Safety-Related Development Assurance in Aerospace ...............77
Figure 25 – Interaction between safety and development process (taken from ARP4754) ....79
Figure 26 – Aircraft development processes........................................................................80
Figure 27 – Security risk assessment activities in the development process V-model (taken from ED202A)..........................................................................................81
Figure 28 – Airworthiness security risk management framework (taken from ED202A) ......82
Figure 29 – Airworthiness Security and Safety Assessment Process in Aerospace (taken from ED202A).................................................................................................84
Figure 30 – The V-Model according to EN 50126 [70] ..........................................................87
Figure 31 – Hourglass Model [71]........................................................................................89
Figure 32 – Risk management process and independent assessment [68].............................91
Figure 33 – Apportionment of functional safety requirements..............................................92
Figure 34 – Scope Exploration related to the Industrial Domain .........................................97
Figure 35 – Hazard Analysis and Risk Assessment related to the Industrial Domain...........98
Figure 36 – Requirements Derivation related to the Industrial Domain...............................99
Figure 37 – System Design related to the Industrial Domain.............................................100
Figure 38 – Hardware and Software Design related to the Industrial Domain....................101
Figure 39 – Development related to the Industrial Domain...............................................102
Figure 40 – Safety and security lifecycle activities ..............................................................104
LIST OF TABLES

Table 1 – Safety integrity levels ................................................................. 16
Table 2 – ISA 99 Security levels ................................................................. 16
Table 3 – Evaluation assurance levels .......................................................... 17
Table 4 – Role Definitions ........................................................................ 37
Table 5 – Scope exploration .................................................................... 40
Table 6 – Hazard and Risk Analysis ............................................................ 41
Table 7 – Threat Analysis ........................................................................ 42
Table 8 – Derive Functional Requirements .................................................. 44
Table 9 – Define System Design ................................................................. 48
Table 10 – Define Hardware/Software Design ............................................. 51
Table 11 – Development .......................................................................... 53
Table 12 – Safety and Security Validation .................................................... 55
Table 13 – Relationship to methods and building blocks .............................. 57
Table 14 – Risk assessment process ............................................................ 66
Table 15 – Security-informed safety case methodology ................................ 67
Table 16 – Phases of EN 50126 lifecycle and mapping to SESAMO generic process ............... 89
Table 17 – SIL Table ................................................................................. 93
Table 18 – The risk tolerability matrix [70]..........................................................94
Table 19 – Overview of safety/risk analysis methods [71]........................................95
Table 20 – Threats/Defences matrix [65] ..............................................................96
EXECUTIVE SUMMARY

This deliverable presents the SESAMO integrated design and evaluation methodology, which aims to combine best practice in safety and security engineering as a unified process.

Almost every standard for critical systems development includes (invariably at the beginning) a section called by various names such as “vocabulary”, “glossary”, or “basic concepts.” Such a section establishes the conceptual and terminological basis for the remainder of the standard. The SESAMO generic process is no different, in the sense that some form of terminological and conceptual foundation must be established as an integral part of the definition of that process. Safety is concerned with protecting the environment from the system whereas security is concerned with protecting the system from the environment. Although each has historically used different terminology, broad equivalences may be established and made operative in the generic process. Other important concepts that have expression in both security and safety include risk and its general treatment under the principle of ALARP; levels to capture risk criticality; techniques and measures for risk reduction; and assurance argumentation.

A major input for the generic process definition was IEC 61508. Its V-Model approach and safety lifecycle have influenced the SESAMO generic process definition. A number of roles are defined that take part in the lifecycle activities of the SESAMO generic process. These roles encompass safety-related activities, security-related activities, combined safety and security activities, and assurance / assessment activities, along with various traditional development activities. The formalized definition of the generic process is accomplished by means of UML activity diagrams. Although the process can be read as a joint process integrating both safety and security activities, the degree of the intermeshing of these activities can be adapted according to the workflow management within an enterprise. The individual steps of the process are described in detail in a tabular format.

The model-based perspective of SESAMO is informed by the technological and methodological background brought into the project by the consortium. Model-based development technology and competence are provided by CHESS in the form of cross-domain component modelling techniques, separation of concerns, and separation of platform-independent and dependent modelling. The model-based perspective is supported by a number of modelling formalisms. One important family of modelling formalisms in SESAMO is based on UML, specifically the SysML profile, which is shared by several of the tools in the SESAMO consortium. Aside from the UML-based family of modelling formalisms, other SESAMO modelling formalisms are associated with specific toolsets, including BDMP, ASCE, and SCYThER. IMACt is another example of the use of a specialized modelling formalism to implement a model-based engineering approach in the avionics domain of SESAMO.

The SESAMO generic process includes a methodology for constructing structured assurance cases for communicating and building confidence in the safety and security properties of a system. Security considerations have a significant impact on various aspects of safety justification. It is necessary to make claims about security properties as well as safety properties, demonstrate compliance to both security and safety standards, and consider a broader set of potential threats and vulnerabilities.
Adaptations of the generic SESAMO process have been explored within several domains, including automotive, avionics, railway, and industrial process control. The adaptations exhibit variations according to specific characteristics of the domain – such as low or high volume (“series”) production.

The definition of the SESAMO methodology is not taking place in a vacuum. Given the nature of the mission-critical systems to be developed with the methodology, an important context exists that cannot be ignored: the standards communities. The safety community in particular has established standards in a number of domains governing safety-related embedded systems development, including avionics, railway, and the automotive industry. A number of security related standards have also been developed.

Given the importance of both safety and security in mission-critical systems, it is natural to consider the inclusion of both within a single standard. However, the integration of safety and security related development processes has been a controversial topic in the standards communities. The SESAMO approach is based on the idea of establishing points of contact between parallel safety and security lifecycle activities. This approach of parallel processes with “weak” trade-off interactions and “strong” interactions for joint activities has the advantage of providing a smooth migration path for the standards communities, allowing them to start with the separate processes of today and gradually identify and implement architectural and process building blocks that promote an ever-closer integration of the processes, while continuously approaching the Holy Grail of a fully integrated SESAMO process.
1 INTRODUCTION

The purpose of this deliverable is to describe the definition of a generic, domain-independent safety and security related design methodology and process suitable for use during the application case studies in WP5.

Inevitably, the integration of safety and security into a single methodology and process has involved the need to confront, analyze, and harmonize the concepts and principles that have developed – often independently – in the safety and security communities. Section 2 provides a discussion of the key results of this work as it affects the approach of the generic process, and effectively serves as an introductory section to the process description itself, setting the context and laying out the fundamental approach.

The generic process is described over several subsections of Section 3. The process is described using a formalized technique of templates and activity descriptions reminiscent of the approach used in AUTOSAR. A set of roles is provided along with the process step descriptions. A key element of the overall process organization is its foreseen use of the building blocks and analysis techniques of WP2 and WP3, and an explicit association of these with individual phases and steps of the generic process is provided.

Following the formalized description of the generic process, two important perspectives on its intended use are provided. In Section 4, a model-based perspective discusses the intended use of tool support for the process through a description of the modelling formalisms currently in use in SESAMO by the various tools provided in the consortium, together with a discussion of potential harmonization issues. Since assurance activities are intended to accompany the development activities across the entire lifecycle, but are not strictly part of the process definition, a comprehensive discussion of the SESAMO approach to these assurance activities is provided in Section 5. It describes in particular the important extension introduced by SESAMO of security-informed safety cases, in comparable detail to the process description itself.

The compatibility of the SESAMO generic process with existing approaches in a number of SESAMO domains is discussed in Section 6. The activity of mapping the generic process to those already in use in those domains has revealed an encouraging level of compatibility, although some concepts and principles of the generic process are expressed in quite different ways in the individual domains, testifying to the challenge of unifying everything under a single umbrella. Most encouraging was a general compatibility of the building blocks and analysis techniques to the approaches of nearly every process studied in those domains.

Section 7 discusses a general approach envisioned for interaction with the various standardization committees in the domains of interest, in order to maximize the take-up of the results of SESAMO in those standards.

Section 8 provides a summary of the overall SESAMO approach in a concise form, highlighting the most important elements of the approach and highlighting the challenges that lie ahead.

Finally, an Appendix provides supplementary supporting material.
2 CONCEPTS AND PRINCIPLES

Almost every standard for critical systems development includes (invariably at the beginning) a section called by various names such as “vocabulary”, “glossary”, or “basic concepts.” Such a section establishes the conceptual and terminological basis for the remainder of the standard. The SESAMO generic process is no different, in the sense that some form of terminological and conceptual foundation must be established as an integral part of the definition of that process. The purpose of this section is to identify the key areas in which harmonization of basic concepts and principles is necessary for the generic process definition.

This cannot be done in a vacuum, of course. In order to create systems that are both safe and secure, we need to ensure that we can deploy the large amount of expertise and work that has been done separately in each domain. The safety and security domains have much in common, but these commonalities are obscured by the use of different concepts and terminologies. To achieve a shared understanding of the key concepts within each domain, we need to establish a lingua franca or even a common ontology. In this chapter we report on some work that is aimed at developing a unified conceptual model for dealing with safety and security.

We first discuss the importance of terminology and taxonomy and review the definitions of security and safety with respect to the dependability taxonomy [30] developed by IFIP WG 10.4. We continue by discussing some more specific aspects of security and safety, namely the need to categorise behaviour and assurance in terms of levels, the concept of security controls, the notion of tolerable risk and the ALARP principle, and the idea of a structured safety case as a means of reasoning and communicating about risk.

Based on our analysis, we conclude that a generic risk identification, analysis, and management life cycle provides a framework around which both safety and security activities can be structured. In order to provide confidence in the effectiveness of this risk management approach, we propose that evidence from the design and evaluation process should be presented in the form of a security-informed safety case that provides a convincing and valid argument that the system is adequately safe and secure for a given application in a given environment.

2.1 TERMINOLOGIES AND TAXONOMIES

It is important to be clear about what we mean by safety and security, particularly if we are attempting to identify what the two domains have in common and how they differ.

The dependability taxonomy presented in [30] provides a way of addressing this issue. According to this taxonomy, safety and security can both be viewed as forms of dependability, which is defined as “the ability of a system to deliver a service that can justifiably be trusted”.

The service delivered by a system is its behaviour as perceived by its users, which might be other systems. Correct service is delivered when the system implements the intended system function. A failure occurs when the delivered service deviates from the correct service. The underlying cause of the failure is called a fault and the manifestation of the fault within the system is called an error. The failure of a component at one level of the system manifests itself as a fault at the next level of the system, so there is a causal chain of fault propagation between different levels of the system.
The dependability taxonomy includes a detailed classification of different kinds of faults and failures, which can be used to explain the distinction between safety and security.

2.1.1 Security terminology

Security is concerned with protecting systems against malicious attacks that seek to compromise the confidentiality, integrity or availability of the system.

Historically, the security community has been distinct from the dependability community and has developed its own terminology, but the most recent version of the dependability taxonomy attempted to document a minimum consensus on the concepts of dependable and secure computing in order to facilitate more technical interaction between the two communities.

In particular, security can be considered to be a form of dependability that focuses on the attributes of confidentiality, integrity and availability in the presence of malicious faults. However, security terminology uses a variety of terms such as attack, vulnerability and intrusion to describe security-related faults, failures and errors.

An intrusion is a malicious interaction fault that compromises the security of a system [31]. This is not the same as an attack because a system can be attacked unsuccessfully. If the system can withstand attack, intrusions can be prevented. In other words, an attack is an intrusion attempt, and an intrusion is the result of a successful attack.

In fact, every intrusion has two underlying causes:

1. a malicious act or attack that attempts to exploit a weakness in the system
2. at least one weakness, flaw or vulnerability in the system that the attacker is able to exploit successfully

Note that vulnerabilities may be introduced during development of a system or during operation. Furthermore, such vulnerabilities can be introduced accidentally or deliberately, with or without malicious intent.

To summarise, the security concepts of attack, vulnerability and intrusion can be expressed in terms of dependability concepts as follows:

- **attack** – a malicious interaction fault, through which an attacker aims to deliberately violate one or more security properties; an intrusion attempt.
- **vulnerability** – a fault created during development of the system, or during operation, which could be exploited to create an intrusion.
- **intrusion** – a malicious, externally-induced fault resulting from an attack that has been successful in exploiting a vulnerability.

2.1.2 Safety terminology

According to the dependability taxonomy, safety is an attribute of dependability. In this sense, safety can be considered to be a particular kind of dependability that focuses on failures of systems that are designed to protect against loss of life or damage to the environment. In this context, it is interesting to note that within the railway community, safety is considered to be one of a set of system
properties that need to be considered, namely reliability, availability, maintainability and safety (RAMS), each of which is a different attribute of dependability.

However, the safety community uses its own terminology for discussing safety-related faults, errors and failures. In particular, it talks about accidents, incidents and hazards. An accident is an undesired event that results in loss or damage whereas an incident is an event that involves no loss (or only minor loss) but has the potential for loss in different circumstances. A hazard is a potential source of harm, a hazardous situation is a situation in which harm is possible, and a hazardous event is an event that may result in harm. A harmful event occurs when a hazardous situation or hazardous event results in harm.

It is also possible to make a distinction between a safe failure and a dangerous failure. If there is a known safe state, the design of the system can be biased towards failing in this direction. This is illustrated in Figure 1, which is taken from the Adelard Safety Case Development Manual [32]:

![Figure 1 – Model of system failure behaviour](image)

**2.1.3 Discussion**

Terminology can be a real source of confusion between different communities and even within the same community, particularly if there is a lack of consensus or clarity about the concepts being discussed. However, although using the same word to mean different things or different words to mean the same thing can cause confusion, it is ultimately the underlying concepts that matter, not the labels that are attached to them.

Safety and security often involve multiple layers of defence and multiple systems, so it is important to be clear about issues such as system boundaries, the scope of each failure, and fault containment; in other words, mechanisms that prevent the failure of one system leading to the failure of another.
Broadly speaking, safety is concerned with protecting the environment from the system whereas security is concerned with protecting the system from the environment. However, since the environment can be viewed as a system in its own right, this suggests a duality or even an equivalence between safety and security – both are concerned with protecting one system from another.

In both safety and security something is being protected and the issue is how to ensure that the protection is adequate. Thus, it is necessary to identify what is being protected, what it is being protected against, what might cause the protection to fail, what the consequences of failure might be, what can be done to reduce those consequences to an acceptable level, and how to determine whether this has been achieved.

Safety is concerned with preventing accidents by identifying potential weaknesses, initiating events, internal hazards and potentially hazardous states and then identifying and applying appropriate mitigations to reduce the risks to a tolerable level. Security is concerned with protecting assets against internal and external threats and vulnerabilities that compromise the asset in some way. Assets are protected using controls that reduce the risk of compromise to an acceptable level.

Juxtaposing these two accounts of safety and security suggests the following equivalences:

- accident = compromise that results in a loss
- weakness = vulnerability
- initiating event = external threat
- hazardous substance (sometimes referred to as just hazard) = internal threat
- hazardous state = compromise
- mitigation = control
- prevention = protection

These equivalences are approximate but indicate the essential similarities between the two disciplines. However, they also highlight some differences in emphasis. For example, although accident and compromise are broadly equivalent, security considers different kinds of compromise. In particular, there is a notion of security attributes: confidentiality, integrity and availability. Accidents are not classified in the same way, although it is easy to conceive of a classification based on what is being harmed (e.g., people, property, environment) as well as the cause of the harm (e.g., loss of integrity). Similarly, security considers compromises to individual assets whereas safety just considers accidents in a broad sense. This reflects the fact that security is concerned with protecting a known set of assets from external threats whereas safety is concerned with preventing a known system from causing harm to an external environment. There are also parallels between safety mitigations and security controls that need to be explored further.

Some aspects of safety and security are domain specific. For example, nuclear safety distinguishes between different kinds of accident (“uncontrolled shutdown”, “release of radiation”, “core meltdown”, “containment breach”). Similarly, military security distinguishes between the consequences of compromising particular assets by grading assets using a classification scheme that reflects their importance (“confidential”, “restricted”, “secret”).
2.1.4 Impact on the SESAMO generic process

The discussion above is relevant to all areas of the SESAMO generic process, from building blocks to activities and analyses. For example, in the joint hazard and threat analysis activity, the approximate equivalences established in this section can be used to help guide the analyses and the resolution of possible conflicts. In some domain-specific instantiations of the generic process, both security and safety related concepts and terminology will already be present (e.g. railway and industrial control), and a mapping and/or resolution from the above discussion can be undertaken. In other domain-specific instantiations (e.g. automotive), concepts and terminology from one or the other area of safety and security will be entirely missing depending on the standard involved, and can be augmented with the concepts and terminology presented here.

2.2 LEVELS AND CLASSIFICATION

In order to make objective statements about safety and security properties, it is useful to have a scale against which these properties can be measured. A particular level of safety or security can then be defined as a point on this scale.

For example, IEC 61508 defines the concept of “safety integrity level” whereas IEC 62443 defines the concept of a “security level”:

\[
\text{safety integrity level} = \text{discrete level (one out of a possible four), corresponding to a range of safety integrity values, where safety integrity level 4 has the highest level of safety integrity and safety integrity level 1 has the lowest}
\]

\[
\text{security level} = \text{level corresponding to the required effectiveness of countermeasures and inherent security properties of devices and systems for a zone or conduit based on assessment of risk for the zone or conduit}
\]

In addition to having a scale for expressing the desired level of safety or security, it is also useful to have a scale for expressing the desired level of confidence in the safety or security of the system. For example, ISO/IEC 15408 defines a (security) evaluation assurance level as:

\[
\text{evaluation assurance level} = \text{set of assurance requirements [...] representing a point on [a] predefined assurance scale}
\]

Before we explore the concept of level in more detail, we first explain these definitions in more detail by summarizing the relevant parts of the standards in question.

2.2.1 Safety integrity levels

IEC 61508 expresses safety levels in terms of a safety integrity requirement:

\[
\text{safety integrity} = \text{the probability of an E/E/PE safety-related system satisfactorily performing the specified safety functions under all the stated conditions within a stated period of time}
\]

Depending on the frequency of demands, the target failure measure is interpreted as either the average probability of dangerous failure on demand (for low demand systems) or the average frequency of dangerous failures (for high demand and continuous demand systems).
IEC 61508 specifies four levels of performance for safety functions. These are called Safety Integrity Levels (SILs) and correspond to a range of safety integrity values. SIL 1 is the lowest level of safety integrity and SIL 4 is the highest level. The range of safety integrity values associated with each SIL is shown in Table 1.

<table>
<thead>
<tr>
<th>Safety integrity level (SIL)</th>
<th>Target failure measure Low demand</th>
<th>Target failure measure High demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>≥10^{-5} to &lt;10^{-4}</td>
<td>≥10^{-9} to &lt;10^{-8}</td>
</tr>
<tr>
<td>3</td>
<td>≥10^{-4} to &lt;10^{-3}</td>
<td>≥10^{-8} to &lt;10^{-7}</td>
</tr>
<tr>
<td>2</td>
<td>≥10^{-3} to &lt;10^{-2}</td>
<td>≥10^{-7} to &lt;10^{-6}</td>
</tr>
<tr>
<td>1</td>
<td>≥10^{-2} to &lt;10^{-1}</td>
<td>≥10^{-6} to &lt;10^{-5}</td>
</tr>
</tbody>
</table>

Table 1 – Safety integrity levels

2.2.2 Security levels

The ISA 99 (IEC 62443) family of standards uses security levels as a qualitative approach to expressing security requirements. As shown in Table 2, there are four different security levels, which are characterised in terms of the threats that they protect against. The definitions of these security levels uses “intentionally vague” terms like “casual”, “coincidental”, “simple”, “sophisticated” and “extended” in order to be generally applicable.

<table>
<thead>
<tr>
<th>Security Level</th>
<th>Level of protection</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Protection against casual or coincidental violation.</td>
</tr>
<tr>
<td>2</td>
<td>Protection against intentional violation using simple means with low resources, generic skills, and low motivation.</td>
</tr>
<tr>
<td>3</td>
<td>Protection against intentional violation using sophisticated means with moderate resources, system specific skills and moderate motivation.</td>
</tr>
<tr>
<td>4</td>
<td>Protection against intentional violation using sophisticated means with extended resources, system specific skills and high motivation.</td>
</tr>
</tbody>
</table>

Table 2 – ISA 99 Security levels

2.2.3 Evaluation assurance levels

The philosophy that underpins the Common Criteria (ISO/IEC 15408) approach to assuring the security of IT systems is that assurance should be based on an evaluation (active investigation) of the IT system that is to be trusted, and that greater assurance arises from the application of greater evaluation effort. Thus, the standard defines a series of Evaluation Assurance Levels (EALs) that are ranked according to the scope, depth and rigour of the evaluation, as shown in Table 3.
### Table 3 – Evaluation assurance levels

<table>
<thead>
<tr>
<th>Evaluation Assurance Level</th>
<th>Nature of evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAL 1</td>
<td>Functionally tested</td>
</tr>
<tr>
<td>EAL 2</td>
<td>Structurally tested</td>
</tr>
<tr>
<td>EAL 3</td>
<td>Methodically tested and checked</td>
</tr>
<tr>
<td>EAL 4</td>
<td>Methodically designed, tested and reviewed</td>
</tr>
<tr>
<td>EAL 5</td>
<td>Semi-formally designed and tested</td>
</tr>
<tr>
<td>EAL 6</td>
<td>Semi-formally verified design and tested</td>
</tr>
<tr>
<td>EAL 7</td>
<td>Formally verified design and tested</td>
</tr>
</tbody>
</table>

## 2.2.4 Discussion

Levels are used widely within safety and security standards in order to make precise statements about safety and security properties. However, it is important to distinguish between the required level of safety or security and the required degree of confidence in that level of safety or security.

Thus, it is helpful to consider two kinds of level: **requirement levels**, which express a safety or security requirement, and **assurance levels**, which express the degree of confidence required in the assessment of a system.

The safety integrity levels defined in IEC 61508 are requirement levels. They are defined in terms of a target failure measure for a safety function that is based on the probability or frequency of dangerous failures. The concept of a security level in IEC 62443 is similar but based on a more subjective and qualitative ranking of security requirements that models the capabilities of the attacker.

In contrast, the evaluation assurance level in IEC 15408 is an assurance level rather than an integrity level. It is actually a requirement on the assurance process rather than the system itself.

Underpinning the idea of a level is an ordinal scale on which safety and security or confidence can be measured – higher levels imply more safety and security or more confidence. This measurement scale can be either quantitative or qualitative. For example, the safety integrity levels of IEC 61508 are based on a target failure measure that is defined as a probability or frequency. In contrast, the security levels of IEC 62443 and the evaluation assurance levels of IEC 15408 are based on a qualitative measure of effort, either the effort expected from the attacker or the effort required from the assessor.

## 2.2.5 Impact on the SESAMO generic process

The generic process has been designed to be fully compatible with the concept of levels, since the major standards of interest also incorporate this concept. However, the generic process remains agnostic whenever possible about certain specific characteristics so that the mapping to specific domains remains feasible. For example, the generic process does not take a position regarding whether integrity levels are quantitative or qualitative, leaving this decision to the domain specific mapping. Nor does the generic process attempt to impose a strict equivalence between safety levels and security levels, preferring only to offer the guidance of the discussion above.
2.3 **RISKS AND LEVELS: TECHNIQUES, MEASURES, AND CONTROLS**

There is clearly a link between risk and level – indeed, risk is also a value that can be measured on an ordinal scale. Safety and security levels arise from the identification of measures that are needed to reduce the risk to a tolerable level. Assurance levels arise from the degree of confidence that is required in the effectiveness of those measures, which is perhaps commensurate with the magnitude of the risk reduction that is being claimed.

Standards often make a link between requirement levels and the methods used to achieve those levels, but this link requires some justification. For example, IEC 61508 specifies the methods that should be used in order to achieve a required level of safety integrity. However, there is no justification for the claim that the application of a particular method will achieve the required level. The standard is simply codifying what is believed to be necessary. This is less of an issue for assurance levels, where it is more defensible to make a link between assurance methods and confidence but similar issues of justification and generality apply.

As noted above, safety-oriented standards like IEC 61508 tend to deal with the issue of risk reduction associated with levels through the specification of so-called techniques and measures (usually in the form of tables). Broadly speaking, security controls are techniques and measures that are used to address security requirements and reduce the risk of a security breach to an acceptable level. There is no consensus about the precise definition of security control, and different standards choose to emphasize different aspects of this broad characterisation.

For example, the NIA glossary [37] uses the following definition of security controls:

*Security controls:* The management, operational, and technical controls (i.e., safeguards or countermeasures) prescribed for an information system to protect the confidentiality, integrity, and availability of the system and its information.

The NIST definition of a security control [38] is similar but differs in two significant ways:

*Security control:* A safeguard or countermeasure prescribed for an information system or an organization designed to protect the confidentiality, integrity, and availability of its information and to meet a set of defined security requirements.

Firstly, NIST do not distinguish between “management, operational and technical” controls, and secondly, NIST make an explicit link between a security control and “a set of defined security requirements”.

In contrast, the ISO 27000 definition of control [33] is based on the concept of managing risk and is rather different in style:

*Control:* means of managing risk, including policies, procedures, guidelines, practices or organizational structures, which can be administrative, technical, management, or legal in nature.

*NOTE:* Control is also used as a synonym for safeguard or countermeasure.

The ISA 99 standard [36] prefers to use the term “countermeasure” to avoid confusion with “control” as in “process control”:
Countermeasure: action, device, procedure, or technique that reduces a threat, a vulnerability, or an attack by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective action can be taken.

This definition focuses on reducing the impact of threats, vulnerabilities and attacks, and is essentially about risk management, although the term “risk” is not used explicitly.

The Common Criteria [35] refer to security functional components rather than controls, but the underlying concept appears to be similar:

“Security functional components express security requirements intended to counter threats in the assumed operational environment of the target of evaluation and/or cover any identified organizational security policies and assumptions.”

In particular, countermeasures are considered to be security objectives for the target of evaluation and are expressed as security functional requirements, formulated in a standardized language defined using security functional components.

2.3.1 Impact on the SESAMO generic process

The general concepts of techniques, measures, and controls as a way of reducing risk to tolerable levels are firmly enshrined in the SESAMO approach of building blocks and analysis methods. The domain-specific mappings presented in Section 6 provide evidence of the validity of this approach and its potential to make a significant contribution to the evolution of techniques, measures, and controls in future versions of relevant standards.

2.4 Tolerability of risk and ALARP

From the preceding discussions it is clear that, at the very highest conceptual level, the SESAMO combined generic process must exhibit a coherent approach to the concept of risk, one that is capable of encompassing both safety and security. The design of high hazard installations and their supporting safety or security systems focuses on minimizing and controlling risks [30]. The cost and rigour of such activities must be proportionate to those risks. Calculations of risk, taking into account severity and likelihood, must demonstrate that the risk has been reduced to a tolerable level.

The “ALARP principle” (the principle that certain risks have to be demonstrated to be “As Low As Reasonably Practicable”) is key to discussing risks and the stopping rules associated with additional design and operational measures. The ALARP principle is based on the assumption that it is possible to compare marginal improvements in safety (marginal risk decreases) with the marginal costs of the risk reduction measures. Safety risks may offer this possibility when they are quantified (e.g., in terms of event probability and of radiation releases), and when the failure rate improvements of the systems controlling the relevant events can be evaluated. Note that the application of the ALARP concept does not necessarily need a quantification of risk reduction. For example, the simple addition of a further safety feature, which costs relatively little, may be obviously worthwhile—qualitative judgements of this nature can often be readily made. Also, marginal does not mean one just considers incremental or small perturbations to the design: sometimes, creative design changes (e.g. substitution of hazardous materials with benign ones) are needed to justify that the risks have been reduced to ALARP.
ALARP found its expression in the well-known “carrot diagram” (see Figure 2 below), which has become the standard means for the exposition of the principle. Levels of risk are divided into three bands, with the width of the wedge representing the level of risk. There are two significant boundaries: the upper one, beyond which risks are not acceptable at all and cannot be justified on any grounds, and a lower one, beyond which risks are considered negligible and no detailed assessment is required. Regulators do not usually require further action to reduce risks unless reasonably practicable measures are available. Within these two boundaries is the ALARP region. At the upper, more risky, end of the ALARP region the risks are only tolerable if costs are judged grossly disproportionate to the risk reduction gained.

![Figure 2 – The ALARP principle](image)

A key part of assessing tolerability is the effective communication of safety and risk. Many sectors with safety-critical systems use the concept of a safety case to facilitate this.

### 2.4.1 Impact on the SESAMO generic process

The generic process incorporates the ALARP principle, which is the most general and widely accepted approach to risk in both the safety and security areas. Specific interpretations arise during mappings to specific domains, such as the so-called Hourglass Principle cited in the railway domain (see Section 6.3.2). The discussion of this section can inform the domain-specific mappings of the treatment of risk. Section 5.1 provides further discussion of the treatment of risk from the perspective of assurance.

### 2.5 SAFETY CASES

Safety cases are an important part of goal-based safety regulation and corporate governance [30]. Explicit safety cases are required for military systems, the off-shore oil industry, rail transport and the nuclear industry.
An early definition of a safety case [40] was

“a documented body of evidence that provides a convincing and valid argument that a system is adequately safe for a given application in a given environment”

Each domain has variants of definitions. For example, a nuclear safety case is defined by the HSE [41] as

“…the totality of documented information and arguments which substantiates the safety of the plant, activity, operation or modification in question. It provides a written demonstration that relevant standards have been met and that risks have been reduced as low as reasonably practicable (ALARP).”

In regulated sectors, the licensee is legally responsible for the safety case. Given the magnitude and complexity of the legislative and technical requirements that have to be met, safety cases have to be structured in a logical manner and be demonstrably adequate.

Safety cases can be seen to support the following [42]:

- **Reasoning and argumentation.** A safety case can be seen as an over-arching framework that allows us to argue whether the claims are substantiated by the evidence. The case might be mainly narrative, using prose to explain the connections between claims and evidence. However cases deal with highly technical subjects and hence they might use specialist notations from the particular disciplines concerned (e.g. from fluid mechanics, computer science). The case will then integrate a selection of technical analyses and other evidence using a formal or graphical notation to show whether the claims have been met; how the evidence is integrated; and the overall structure of the case and the thrust of the argument.

- **Negotiation, communication, and trust.** The safety case represents a boundary object between the different stakeholders who have to agree (or not) on the claims being made about the system. To this end it has to be detailed and rigorous enough to effectively communicate the case and allow challenges and the subsequent deepening of the case. A safety case has to support an argument that the requirements placed upon a system are met. As such, the safety case contains claims about the properties of the system and, following a systematic approach, has arguments that demonstrate that these claims are substantiated or rebutted by evidence.

Thus, to implement a safety case we need to

- make an explicit set of claims about the system
- produce the supporting evidence
- provide a set of arguments that link the claims to the evidence
- make clear the assumptions and judgements underlying the arguments
- allow different viewpoints and levels of detail

Current safety case practice makes use of the basic approach developed by Toulmin [43] where claims are supported by evidence and a “warrant” or argument that links the evidence to the claim, as shown in Figure 3. There are variants of this basic approach that present the claim structure
graphically such as Goal Structuring Notation (GSN) [44] or Claims-Arguments-Evidence (CAE) [30].

![Figure 3 – Toulmin’s formulation of a claim](image)

There are several different ways of constructing such a justification. The three main approaches can be characterized in terms of a safety justification “triangle” [8]:

- Claims about the system’s safety behaviour (positive properties).
- The use of accepted standards and guidelines.
- Analysis of potential vulnerabilities (negative properties).

The first approach is goal-based—where specific safety claims for the systems are supported by arguments and evidence at progressively more detailed levels. The second approach is based on demonstrating compliance to known safety standards. The final approach is a vulnerability-based argument where it is demonstrated that potential vulnerabilities within a system do not constitute a problem—this is essentially a “bottom-up” approach as opposed to the “top-down” approach used in goal-based methods. These approaches are not mutually exclusive, and a combination of approaches can be used to support a safety justification, especially where the system consists of both off-the-shelf (OTS) components and application-specific elements.

Security considerations have an impact on each aspect of the safety justification triangle. It is necessary to make claims about security properties as well as safety properties, demonstrate compliance to both security and safety standards, and consider a broader set of potential threats and vulnerabilities. The hazards remain the same but the judgments we make about the likelihood of a hazard leading to an accident might be different because we are no longer dealing with a benevolent threat model.

To summarise, the motivation for a safety case is to

- provide an assurance viewpoint – for efficient review
- provide a focus and rationale for activities – leading to efficient analysis and evaluation
- provide a reviewable approach – so that all stakeholders can be involved
- demonstrate the discharge of duty to the public and shareholders
- allow interworking between standards and innovation
The emphasis should be on the behaviour of the product and not just the process used to develop it: a useful slogan is "What has been achieved not how hard you have tried".

2.5.1 Impact on the SESAMO generic process

The concept of a security-informed safety case is fully embraced by the SESAMO generic process, and considered an important extension to traditional processes and means of harmonizing approaches to argumentation of dependability factors during assessment procedures. A methodology for developing the security-informed safety case has been developed for inclusion into the generic process, and is described and documented over a number of subsections of Section 5.

Domain-specific mappings are more or less accommodating of this approach to assurance. For example, in the automotive domain, there is no published work yet on arguing security along with safety, although the issue is being considered. But in the railway domain, an embryonic approach to security and a basis for combined argumentation of safety and security exists – it is the intention of the SESAMO generic process to support the fuller development of this approach as the standards evolve to support both safety and security, as discussed in Chapter 7.
3 SESAMO GENERIC PROCESS

3.1 INTRODUCTION

The SESAMO generic process will be based on the principles of risk management outlined in ISO 27005 [5]. Figure 4 provides a high level view of such a process:

![Figure 4 – Risk management process (from ISO 27005)](image)

Having established the system context, the first step is to perform a risk assessment, which involves identifying, analysing and evaluating the risks associated with the system. This is followed by a process of risk treatment, in which decisions are taken about how best to deal with each risk.

Both risk assessment and risk treatment can be iterative, the goal being to reduce the residual risks to an acceptable level, as shown in Figure 5, which shows the risk treatment process in more detail.
Figure 5 – Risk treatment activity (from ISO 27005)

There are four basic approaches to risk treatment:

- **Risk modification**: manage the level of risk by introducing appropriate controls
- **Risk retention**: make an objective and informed decision to accept the risk
- **Risk avoidance**: modify the system to avoid the condition that gives rise to the risk
- **Risk sharing**: pass some or all of the risk onto some other system

These choices need to be recorded as part of the safety case for the system.
Furthermore in order to define the generic SESAMO process it is obvious to take already standardized and existing process definitions in the area of safety and security into considerations. Thus a major input for the generic process definition was taken from the mother standard of nearly all existing safety standards applied in the different domains, IEC 61508. Its V-Model approach and its safety lifecycle influenced the SESAMO process definition. The following section give a rough overview of this lifecycle. This should be read as a basis for the upcoming joint safety and security process definition in chapter 3.4.

### 3.2 SAFETY LIFECYCLE (IEC 61508)

The following definition of a Safety Lifecycle is based on the IEC 61508 standard (see Figure 6).

![Safety Lifecycle Diagram](image)

**Figure 6 – Safety Lifecycle – Overview.**

#### 3.2.1 Safety Lifecycle Overview

The objectives of the main phases of the overall safety lifecycle are described below:

- **Concept Phase.** To develop a sufficient level of understanding for the equipment under control (EUC) and its environment to enable the other safety lifecycle activities to be carried out satisfactorily.

- **Overall Scope Definition.** To determine the boundary between the EUC and EUC control system (ECS), and to specify the scope of the hazard and risk analysis.
• **Hazard and Risk Analysis.** To determine the hazards relating to the EUC and ECS for all reasonably foreseeable circumstances, and to determine the EUC risks associated with the hazardous events.

• **Overall Safety Requirements.** To determine the target safety integrity requirements and the safety function requirements.

• **Overall Safety Requirements Allocation.** To allocate the safety functions to designated safety related systems and other risk reduction measures.

• **System Safety Requirements Specification.** To define the system safety requirements, in terms of system safety function requirements and system safety integrity requirements, in order to achieve the required functional safety.

• **Safety-Related Systems Realization.** To create safety-related systems (SRS) conforming to the specification for the system safety requirements, including system safety function requirements and system safety integrity requirements.

For clarification, the following gives a brief introduction into the objectives and outputs of each corresponding phase.

### 3.2.2 Concept Phase

**Objectives**

- Describe the system with focus on
  
  - EUC
  - Environment
  - Control Functions

- Make first considerations about hazards:
  
  - Identify possible sources of hazards
  - Collect Information about known hazards and safety regulations
  - Consider hazards resulting from interaction with other systems

**Output**

- Documentation that…
  
  - Enables an easy understanding of EUC/Environment/Control Functions concept
  - Provides information about probable preliminary hazards and their implications regarding system safety and safety regulations
3.2.3 Overall Scope Definition

Objectives

➢ Define system boundary and relation between EUC and ECS
➢ Define the scope for the following Hazard and Risk Analysis:
   • Physical Equipment
   • External Events
   • Equipment and systems that are associated with hazards

Output

➢ Documentation:
   • Relates EUC and ECS and defines the system boundaries with respect to preliminary identified hazards

3.2.4 Hazard and Risk Analysis

Objectives

➢ Determine hazards and risks for the EUC
   • Organize (independent) assistance personal
   • Identify hazards for EUC and ECS
   • Establish a risk matrix through risk analysis
   • Identify risk reduction measures (RRM)
   • Check if the already taken risk reduction measures (RRM) are sufficient (ALARP – as low as reasonably practicable)

Output

➢ Safety Report containing:
   • Hazard scenarios (with consideration of foreseeable scenarios, event sequences leading to hazards, and misuse) containing consequences and likelihood of hazardous events
   • Risk and safety level classification matrix
   • Risk reduction measurement
3.2.5 Overall Safety Requirements

Objectives

- Implement a safety specification
  - Specify overall safety function requirements (functionally)
  - Specify target safety integrity requirements for each safety function (e.g. required risk reduction)
  - Specify overall safety integrity requirements
  - Determine SRS (included in ECU or separate SRS) or other RRMs
  - Assign SILs to safety functions

Output

- Safety specification with:
  - Safety function requirements (independent from the implementation technology)
  - Safety integrity requirements
  - Table with SILs assigned to safety functions

3.2.6 Overall Safety Requirements Allocation

Objectives

- Map overall safety function requirements onto designated SRSs
  - Specify the designated SRSs
  - Map safety functions requirements
  - Specify safety integrity requirements for each safety function
  - Map safety integrity requirements
  - Allocate target failure measure and an associated SIL to each safety function

Output

- Safety report containing:
  - Safety function allocation diagram showing the relationship between hazards, safety functions, safety function requirements, SILs and the allocation of safety functions onto SRS or other RRMs
  - Table with target failure rate and SIL allocated to safety functions
3.2.7 Overall Safety Requirements Specification

Objectives

- Define system safety requirements
  - Define system safety function requirements
  - Define system safety integrity requirements

Output

- System Safety Requirements Specification containing:
  - Requirements for safety functions (with associated SILs) fit for development
    - Time and data interfaces
    - Operation modes
    - Information relevant for system safety
  - Requirements for safety integrity
    - SIL for each safety function
    - Operation mode for each safety function
    - Required duty-cycle, lifetime
    - Possible environmental conditions
    - Constraint and limiting conditions

3.2.8 Safety-Related Systems Realization

Objectives

- Implement SRS conforming to specification
- Implement the SRS based on
  - System safety function requirements
  - System safety integrity requirements

Output

- SRS Realization
- Implementation documentation
A graphical overview of the shown Industrial Domain Safety Lifecycle is depicted in the following Figure 7.

### Concept

**Describe the system with focus on:**
- EUC
- Environment
- Control Functions

**Make first considerations about hazards:**
- Identify possible sources of hazards
- Collect Information about known hazards and safety regulations
- Consider hazards resulting from interaction with other systems

### Overall Scope Definition

**Define system boundary and relation between EUC and ECS**

**Define the scope for the following Hazard and Risk Analysis**
- Physical Equipment
- External Events
- Equipment and systems that are associated with hazards

### Hazard and Risk Analysis

**Determine hazards and risks for the EUC**
- Organize (independent) assistance personal
- Identify hazards for EUC and ECS
- Establish a risk matrix through risk analysis
- Identify risk reduction measures (RRM)
- Check if the already taken RRM are sufficient (ALARP)

**Output:**

**Safety Report containing:**
- Hazard scenarios (with consideration of foreseeable scenarios, event sequences leading to hazards, and misuse) containing consequences and likelihood of hazardous events
- Risk and safety level classification matrix
- Risk reduction measures (SRS)

### Overall Safety Requirements

**Implement a safety specification**
- Specify overall safety function requirements (functionally)
- Specify target safety integrity requirements for each safety function (e.g. required risk reduction)
- Specify overall safety integrity requirements
- Determine SRS (included in ECU or separate SRS) or other RRM
- Assign SILs to safety functions

**Output:**

**Safety specification with:**
- Safety function requirements (independent from the implementation technology)
- Safety integrity requirements
- Table with SILs assigned to safety functions
3.3 **ROLE DEFINITIONS**

The following table defines the roles that take part in the SESAMO process’ activities.

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product Lifecycle Manager</strong>&lt;br&gt;<strong>PLM</strong>&lt;br&gt;The PLM is responsible to organize all activities related to the product, starting from the product concept phase unto the release of the product.</td>
<td>• Product objectives&lt;br&gt;• Technical objectives&lt;br&gt;• Quality objectives&lt;br&gt;• Economic and risk objectives&lt;br&gt;• Scheduling objectives&lt;br&gt;• Safety and security objectives&lt;br&gt;• Participates in safety and security interference analysis&lt;br&gt;• Participates in the choosing of the implementation platform&lt;br&gt;• Participates in functional safety and security assessment&lt;br&gt;• Validates goals and achievements</td>
</tr>
<tr>
<td>Role</td>
<td>Responsibilities</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>System Architect</strong></td>
<td>The System Architect has the task to design the architecture of the system, which includes the design and designation of components and interfaces with focus on reasonable usability and compatibility.</td>
</tr>
<tr>
<td></td>
<td>• Component architecture design</td>
</tr>
<tr>
<td></td>
<td>• Compatibility of components</td>
</tr>
<tr>
<td></td>
<td>• Platform definition and development including interfaces</td>
</tr>
<tr>
<td></td>
<td>• Platform configuration and change management</td>
</tr>
<tr>
<td></td>
<td>• Requirement specifications</td>
</tr>
<tr>
<td></td>
<td>• Takes part in interference analysis concerning safety and security influences</td>
</tr>
<tr>
<td><strong>Security Manager</strong></td>
<td>The Security Manager is responsible for enforcing security into the design with respect to the interests of the organization.</td>
</tr>
<tr>
<td></td>
<td>• Checks and provides state-of-the-art security standards and knowledge</td>
</tr>
<tr>
<td></td>
<td>• Tracks security requirements and keeps track of the interference of security requirements on safety</td>
</tr>
<tr>
<td></td>
<td>• Selects team members</td>
</tr>
<tr>
<td></td>
<td>• Defines processes for the procurement of security-related components</td>
</tr>
<tr>
<td></td>
<td>• Coordination of activities with the Product Lifecycle Manager, System Architect, Safety Manager, Quality Manager, Procurement Manager</td>
</tr>
<tr>
<td></td>
<td>• Driving the security activities in the &quot;Concept of Operations&quot; phase</td>
</tr>
<tr>
<td></td>
<td>• The identification security violations</td>
</tr>
<tr>
<td></td>
<td>• Threat analysis</td>
</tr>
<tr>
<td></td>
<td>• Derive functional security requirements</td>
</tr>
<tr>
<td></td>
<td>• Taking part in interference analysis activities concerning safety and security requirements</td>
</tr>
<tr>
<td></td>
<td>• In the System Design phase</td>
</tr>
<tr>
<td></td>
<td>• Derives technical security requirements and HW/SW security requirements</td>
</tr>
<tr>
<td></td>
<td>• Realization and documentation of designed security measures</td>
</tr>
<tr>
<td></td>
<td>• Selects security-related BBs</td>
</tr>
<tr>
<td></td>
<td>• Carries out interference analysis</td>
</tr>
<tr>
<td></td>
<td>• Development phase:</td>
</tr>
<tr>
<td></td>
<td>• Takes part in choosing a security-conform implementation platform</td>
</tr>
<tr>
<td></td>
<td>• Functional Security and Safety Assessment phase:</td>
</tr>
<tr>
<td></td>
<td>• checks if the security requirements are met by the current implementation</td>
</tr>
<tr>
<td><strong>Role</strong></td>
<td><strong>Responsibilities</strong></td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Safety Manager</strong></td>
<td></td>
</tr>
</tbody>
</table>
| The Safety Manager carries out several activities of the safety lifecycle (in line with the corresponding safety standards, such as the IEC 61508) starting from the concept phase, to the derivation of safety requirements, down to the functional safety assessment. | - Checks if all activities are carried out in line with the standard IEC 61508  
- Tracks all safety requirements and liaises with the security manager to resolve conflicts between safety and security requirements  
- Organizes current safety standards / communicates with safety authorities  
- Selects team members  
- Defines processes for the procurement of safety-related components  
- Coordination of activities with the Product Lifecycle Manager, System Architect, Security Manager, Quality Manager, Procurement Manager  
- Drives safety activities in the "Concept of Operations" phase  
  - The identification of functions and malfunctions  
  - Hazard Analysis and Risk Assessment  
  - Derive Functional Requirements  
  - Taking part in interference analysis activities concerning safety and security requirements  
- In the System Design phase:  
  - Derives technical safety requirements and HW/SW safety requirements  
  - Realization and documentation of system reliability diagrams  
  - Selects safety-related BBs  
  - Carries out interference analysis  
- Development phase:  
  - Takes part in choosing a safety-conform implementation platform  
- Functional Security and Safety Assessment phase:  
  - checks if the safety requirements are met by the current implementation |
<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibilities</th>
</tr>
</thead>
</table>
| **Quality Manager**      | • Enforces quality interests with respect to other project stakeholders  
                          • Contributes to the quality requirements definition  
                          • Continuous involvement in the enhancement of processes and products  
                          • Reports quality states and consults project lifecycle manager  
                          • Validates product and process quality requirements  
                          • Creates quality plans  
                          • Controls, supports and documents quality processes |
| **Verification Manager** | • Plans and controls system verification activities  
                          • Supports test activities during development  
                          • Approves product release concerning quality  
                          • Enforces adequate documentation quality  
                          • Ensures adherence of safety concepts  
                          • Defines test cases  
                          • Test activity planning  
                          • Monitors verification progresses  
                          • Validates functionality of product features  
                          • Revises involvement with certification authorities in case of safety-related systems |
| **Software Design Engineer** | • Participates in deriving technical safety and security requirements  
                          • Participates in deriving software safety and security requirements  
                          • Participates in the application/mapping of safety and security building blocks  
                          • Participates in safety and security interference analysis  
                          • Software design  
                          • Participates in planning development activities including work and scope assessments  
                          • Guarantees consistency of work artefacts  
                          • Executes quality measures (e.g. code reviews)  
                          • Maintains documentation  
                          • Participates in software tests and integration  
                          • Participates in functional safety and security assessment |
<p>| <strong>Software Design Engineer</strong> | The Software Design Engineer executes software design with respect to carefully designed interfaces. |</p>
<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibilities</th>
</tr>
</thead>
</table>
| **Hardware Design Engineer**   | • Hardware design  
• Participates in deriving technical safety and security requirements  
• Participates in deriving hardware safety and security requirements  
• Participates in the application/mapping of safety and security building blocks  
• Participates in safety and security interference analysis  
• Participates in hardware development plan  
• Participates in integration and test phase  
• Participates in functional safety and security assessment |
| **Software Verification Engineer** | • Plans tests for software modules and software integration  
• Determines test methodology together with suitable tools  
• Creates and maintains software test cases  
• Executes software tests and integration tests  
• Documents and distributes test results  
• Validates safety and security  
• Assesses functional safety and security |
| **Hardware Verification Engineer** | • Plans hardware tests  
• Determines hardware test methodology and chooses suitable test tools  
• Creates and maintains hardware test cases  
• Executes hardware tests and integration tests  
• Evaluates and distributes hardware test results  
• Validates Safety and security  
• Assesses functional safety and security |
| **System Test Engineer**       | • Plans system test activities  
• Creates and maintains system test cases  
• Checks for fulfilment of system requirements  
• Elaborates test cases with conformity to safety standards  
• Participates in system tests  
• Documents system test results (also for the use by certification authorities)  
• Validates safety and security  
• Assesses functional safety and security |
<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibilities</th>
</tr>
</thead>
</table>
| **Procurement Manager** | • Checks and evaluates suppliers regarding safety and security requirements for components  
|                       | • Follows processes and guidelines provided from the safety manager and the security manager for procurement  
|                       | • Looks for the economically optimal solution available on the market  
|                       | • Contract management  
|                       | • Assesses component availability, quality, and trustworthiness  
|                       | • Participates in the identification of functions  
|                       | • Participates in the choosing of the implementation platform  
|                       | • Participates in the mapping of building blocks onto the implementation platform  
|                       | • Participates in the functional security and safety assessment  |
| **Independent Assessor** | • Gives independent and genuine advice to safety and security related interference activities  
|                       | • Objectively advises the functional safety and security assessment of the design  
|                       | • Creates documentation of all taken design measures and trade-offs taken between safety and security design characteristics  |

**Table 4 – Role Definitions**

### 3.4 FORMALIZED GENERIC PROCESS DEFINITION

This section formalizes the generic process definition of SESAMO using UML activity diagrams. The overall process is based on the V-Model approach as depicted in Figure 8 and adopts ideas from IEC 61508 and ISO 26262. The different steps in this process will be described in detail in the following sub sections, with the main emphasis put on the activities related to safety and security. Because of the fact that this process is closely related to already existing safety standards, the extensions made for security are marked in red on the UML activity diagrams.

Although the SESAMO generic process can be read as a joint process integrating both safety and security activities, the degree to which these activities are intermeshed can be adapted according to the workflow management within an enterprise. That means depending on the organizational structure of an enterprise the activities can be performed in parallel within one team consisting of both security and safety experts as well as in more separated or sequential ways involving different teams. However there have to be well-defined synchronization points between the security and safety activities that should lead to common work products, e.g. a common system design. This issue is discussed in more detail in Chapter 7.

The description of the activities includes their associated roles and corresponding work products.
3.4.1 Concept of Operations

In this step a shared understanding of the system has to be developed amongst the stakeholders and how it will be operated and maintained. The Concept of Operations is documented to provide a foundation for more detailed analyses that will follow. It will produce a set of functional requirements including requirements related to safety and security together with a preliminary architecture of the system (see Figure 9).

Figure 8 – Generic Process Definition
Figure 9 – Concept of Operation

**Activity**  | **Scope Exploration**
--- | ---
**Description** | The objective is to define and describe the system to be developed, its dependencies on, and interaction with, the environment and other systems. The second objective is to support an adequate understanding of the system so that the activities in subsequent phases can be performed.

**Work Product Kind**  | **Name**  | **Note**
--- | --- | ---
Consumes |  |  |
Produces | Functional Description | describing the purpose and functionality of the system, including the operating modes and states; the operational and environmental constraints; legal requirements; behaviour achieved by similar systems; assumptions on behaviour expected from the system; boundary of the system, its interfaces, and the assumptions concerning its interaction with other systems
Malfunctions | A set of malfunctioning behaviours that have to be considered with respect to safety
### Table 5 – Scope exploration

<table>
<thead>
<tr>
<th>Sub-Activities</th>
<th>Description</th>
<th>Assigned Roles</th>
</tr>
</thead>
</table>
| Identify Functions   | functional and non-functional requirements of the system as well as the dependencies between the system and its environment shall be made available | Product Lifecycle Manager  
System Architect  
Security Manager  
Safety Manager  
Quality Manager  
Procurement Manager |
| Identify Malfunctions | malfunctioning behaviour is identified for all functions of the system                               | Product Lifecycle Manager  
System Architect  
Safety Manager  
Quality Manager |
| Identify Security Violations | potential threats are identified                                                                       | Product Lifecycle Manager  
System Architect  
Security Manager  
Quality Manager |

### Activity Hazard and Risk Analysis

**Description**

The objective of the hazard and risk analysis is to identify and categorize the hazards that malfunctions in the system can trigger and to formulate the safety goals related to the prevention or mitigation of these hazards, in order to avoid unreasonable risk.

**Work Product Kind**

<table>
<thead>
<tr>
<th>Name</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumes Functional Description</td>
<td></td>
</tr>
<tr>
<td>Malfunctions</td>
<td></td>
</tr>
<tr>
<td>Operational Situation</td>
<td>a catalogue of situations and conditions under which the system is supposed to be used</td>
</tr>
<tr>
<td>Produces Hazard analysis and risk assessment</td>
<td>lists the malfunctions under well-defined and relevant operational situations and determines for every resulting hazard its criticality level which in-</td>
</tr>
</tbody>
</table>
includes parameters for the severity, the exposure and the controllability of the hazard

| Safety Goals | a set of top-level safety requirements derived from every hazard |
| Verification review report of the hazard analysis and risk assessment and the safety goals | this report shows: completeness with regard to operational situations and hazards; compliance with the system’s description; consistency with related hazard analyses and risk assessments; completeness of the coverage of the hazards; consistency of the assigned criticality levels with the corresponding hazards |

<table>
<thead>
<tr>
<th><strong>Sub-Activities</strong></th>
<th><strong>Description</strong></th>
<th><strong>Assigned Roles</strong></th>
</tr>
</thead>
</table>
| Define Hazards    | relate malfunctioning behaviour of the system with relevant operations situations and define the potential effects as hazards | System Architect  
Safety Manager  
Quality Manager |
| Determine Safety Criticality Level | assess the hazard in respect to its severity, its controllability and exposure | System Architect  
Safety Manager  
Quality Manager |
| Derive Safety Goals | derive top-level safety requirements out of the hazard | System Architect  
Safety Manager  
Quality Manager |

Table 6 – Hazard and Risk Analysis

<table>
<thead>
<tr>
<th><strong>Activity</strong></th>
<th><strong>Threat Analysis</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>The objective of the threat analysis is to identify and to categorize the potential security violations that might cause malfunctioning behaviour in the system and to formulate the security goals related to the prevention or mitigation of the threats, in order to avoid unreasonable risk.</td>
</tr>
<tr>
<td>Work Product Kind</td>
<td>Name</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Consumes</td>
<td>Functional Description</td>
</tr>
<tr>
<td></td>
<td>Security Violations</td>
</tr>
<tr>
<td></td>
<td>Operational Situations</td>
</tr>
<tr>
<td>Produces</td>
<td>Threat analysis</td>
</tr>
<tr>
<td></td>
<td>Security Goals</td>
</tr>
<tr>
<td></td>
<td>Verification review report of the threat analysis and the security goals</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sub-Activities</th>
<th>Description</th>
<th>Assigned Roles</th>
</tr>
</thead>
</table>
| Define Threats         | consider security violations of the system in relevant operational situations and define the potential threats | System Architect  
                        |                                                                             | Security Manager  
                        |                                                                             | Quality Manager     |
| Determine Security Criticality Level | assess the threats in respect to the need resources and needed skills of an attacker | System Architect  
                        |                                                                             | Security Manager  
                        |                                                                             | Quality Manager     |
| Derive Security Goals  | derive top-level security requirements out of the hazard                      | System Architect  
                        |                                                                             | Security Manager  
                        |                                                                             | Quality Manager     |

Table 7 – Threat Analysis
### Activity

**Derive Functional Requirements**

### Description

The objective of this activity is to derive the functional safety and security requirements from the safety goals and the security goals, and to allocate them to a preliminary architecture of the system or to external measures. This also already involves the selection of the necessary building blocks defined by WP2.

### Work Product Kind

<table>
<thead>
<tr>
<th>Name</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Consumes</strong></td>
<td></td>
</tr>
<tr>
<td>Functional Description</td>
<td></td>
</tr>
<tr>
<td>Hazard analysis and risk assessment</td>
<td></td>
</tr>
<tr>
<td>Threat analysis</td>
<td></td>
</tr>
<tr>
<td>Architecture Assumptions</td>
<td></td>
</tr>
<tr>
<td>Safety Goals</td>
<td></td>
</tr>
<tr>
<td>Security Goals</td>
<td></td>
</tr>
<tr>
<td><strong>Produces</strong></td>
<td></td>
</tr>
<tr>
<td>Functional Safety Requirements</td>
<td>address the usage of the safety related building blocks which deal with fault detection, failure mitigation and transitioning to a safe state</td>
</tr>
<tr>
<td>Functional Security Requirements</td>
<td>address the usage of the security related building blocks</td>
</tr>
<tr>
<td>Preliminary Safety/Security Architecture</td>
<td>a first functional architecture showing the used building blocks</td>
</tr>
<tr>
<td>Verification report</td>
<td>this report shows: completeness and maturity of the functional requirements and coverage of the functional requirements in the preliminary architecture</td>
</tr>
<tr>
<td>Validation plan</td>
<td>initial plan on how to validate the functional requirements</td>
</tr>
</tbody>
</table>
### Sub-Activities

<table>
<thead>
<tr>
<th>Description</th>
<th>Assigned Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Derive Functional Safety Requirements</strong></td>
<td>System Architect</td>
</tr>
<tr>
<td></td>
<td>Safety Manager</td>
</tr>
<tr>
<td></td>
<td>Quality Manager</td>
</tr>
<tr>
<td>starting from the safety goals functional safety requirements are derived</td>
<td></td>
</tr>
<tr>
<td><strong>Derive Functional Security Requirements</strong></td>
<td>System Architect</td>
</tr>
<tr>
<td></td>
<td>Security Manager</td>
</tr>
<tr>
<td></td>
<td>Quality Manager</td>
</tr>
<tr>
<td>starting from the security goals functional security requirements are derived (EVITA [11] defines a security requirements management process for the automotive domain which can be applied here)</td>
<td></td>
</tr>
<tr>
<td><strong>Interference Analysis</strong></td>
<td>Product Lifecycle Manager</td>
</tr>
<tr>
<td></td>
<td>System Architect</td>
</tr>
<tr>
<td></td>
<td>Security Manager</td>
</tr>
<tr>
<td></td>
<td>Safety Manager</td>
</tr>
<tr>
<td></td>
<td>Quality Manager</td>
</tr>
<tr>
<td></td>
<td>Independent Assessor</td>
</tr>
<tr>
<td>first analyses take place in order to find out whether trade-offs or synergies can be identified between safety and security requirements and their corresponding building blocks</td>
<td></td>
</tr>
</tbody>
</table>

Table 8 – Derive Functional Requirements

#### 3.4.2 System Design

The objective of this activity is to develop the system design and the technical safety and security concept that corresponds with the functional requirements and the technical safety and security requirements specification of the system.

The development of the system design and the technical safety/security concept is based on the technical safety and security requirements specification derived from the functional safety/security concept. The different sub-activities can be applied iteratively. In particular, in the interference analysis (see Figure 11) the design and the evaluation should be interleaved, so that evidence from design informs the evaluation and the results of evaluation inform design. This is illustrated in Figure 10, which shows the interaction between the design activities based on threat/risk assessment, and an evaluation activity, based on security assurance.
Figure 10 – Interleaved activities

The technical safety and security requirements and safe/secure system design that result from this phase are based on an interference analysis of the safety and security building blocks that are available to satisfy the functional safety and security requirements.
### Figure 11 – System Design

<table>
<thead>
<tr>
<th>Activity</th>
<th>Define System Design</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Work Product Kind</strong></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>Consumes</td>
<td>Functional Description</td>
</tr>
<tr>
<td></td>
<td>Hazard analysis and risk assessment</td>
</tr>
<tr>
<td></td>
<td>Threat analysis</td>
</tr>
<tr>
<td>Sub-Activities</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Derive Technical Safety Re-</td>
<td>based on the functional safety requirements technical safety requirements are derived</td>
</tr>
<tr>
<td>quirements</td>
<td></td>
</tr>
<tr>
<td>Apply Safety Building Blocks</td>
<td>according to the technical safety requirements the corresponding safety related building blocks have to be incorporated into the system design</td>
</tr>
</tbody>
</table>

Functional Safety Requirements: address the usage of the safety related building blocks which deal with fault detection, failure mitigation and transitioning to a safe state.

Functional Security Requirements: address the usage of the security related building blocks.

Preliminary Safety/Security Architecture: a first functional architecture showing the used building blocks.

Technical Safety Requirements: Produces.

Technical Security Requirements: Produces.

Safe/Secure System Design: includes the built-in safety and security building blocks derived from the technical requirements.

System verification report: shows the: compliance and consistency with the functional concept; and compliance with the preliminary architectural design assumptions.

Validation plan (refined): validation plan has to be refined according to the technical requirements.
| Derive Technical Security Requirements | based on the functional security requirements technical security requirements are derived | Security Manager  
Quality Manager  
SW Design Engineer  
HW Design Engineer |
|---|---|---|
| Apply Security Building Blocks | according to the technical security requirements the corresponding security related building blocks have to be incorporated into the system design | Security Manager  
Quality Manager  
SW Design Engineer  
HW Design Engineer |
| Interference Analysis | the interference analysis will use methods described in WP3 (see also section 3.5) in order to find trade-offs and synergies between the used building blocks; this might lead to new requirements which in the end will have impact on the system design again | System Architect  
Security Manager  
Safety Manager  
SW Design Engineer  
HW Design Engineer  
Quality Manager  
Independent Assessor |

**Table 9 – Define System Design**

Similar activities are going on when designing the hardware and software architecture (see Figure 12).
Figure 12 – Hardware and Software Design

<table>
<thead>
<tr>
<th>Activity</th>
<th>Define Hardware/Software Design</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Work Product Kind</strong></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>Consumes</td>
<td>Technical Safety Requirements</td>
</tr>
<tr>
<td></td>
<td>Technical Security Requirements</td>
</tr>
<tr>
<td></td>
<td>System Design</td>
</tr>
<tr>
<td>Produces</td>
<td>HW/SW Safety Requirements</td>
</tr>
<tr>
<td>Sub-Activities</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Derive HW/SW Safety Requirements</td>
<td>based on the technical safety requirements</td>
</tr>
<tr>
<td></td>
<td>HW/SW safety requirements are derived</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Apply Safety Building Blocks</td>
<td>according to the HW/SW safety requirements the corresponding safety related building blocks have to be incorporated into the HW/SW Architecture</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Derive HW/SW Security Requirements</td>
<td>based on the technical security requirements</td>
</tr>
<tr>
<td></td>
<td>HW/SW security requirements are derived</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Apply Security Building Blocks</td>
<td>according to the HW/SW security requirements the corresponding security related building blocks have to be incorporated into the HW/SW Architecture</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.4.3 Development

The objective of this activity is to implement the system according to the technical design specification resulting from the step before (see section 3.4.2), especially the building blocks that guarantee a safe and secure system. Depending on the chosen approach and the target platform configuration data and code are derived from the models either automatically or manually.
### Figure 13 – Development

<table>
<thead>
<tr>
<th>Activity</th>
<th>Development</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Work Product Kind</strong></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>Consumes</td>
<td>HW/SW Safety Requirements</td>
</tr>
<tr>
<td></td>
<td>HW/SW Security Requirements</td>
</tr>
<tr>
<td></td>
<td>HW/SW Architecture</td>
</tr>
<tr>
<td>Sub-Activities</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Choose Implementation Platform</td>
<td>the choice might depend on the target domain, for instance Avionics, Automotive (section 6.1.2.3 and Annex A), etc.</td>
</tr>
<tr>
<td>Map modelled building blocks to platform specific realizations</td>
<td>if building block implementations are provided by the chosen platform the corresponding requirements from the technical concept have to be traced here (see also section 6.1.2.3 and Annex A as an example)</td>
</tr>
<tr>
<td>Derive configuration information</td>
<td>based on the requirements building blocks have to be parameterized</td>
</tr>
<tr>
<td>Generate/Provide code</td>
<td>implementation is provided</td>
</tr>
</tbody>
</table>

Table 11 – Development
### 3.4.4 Safety and Security Validation

The goal of this activity is to validate fulfilment of functional safety and security requirements (see Figure 14). In many cases similar requirements need to be validated from both safety and security perspective. Then, it is economic to share the same validation infrastructure for both cases.

![Figure 14 – Safety and Security Validation](image)

<table>
<thead>
<tr>
<th>Activity</th>
<th>Safety and Security Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work Product Kind</td>
<td>Name</td>
</tr>
<tr>
<td>Consumes</td>
<td>Functional Safety Requirements</td>
</tr>
<tr>
<td></td>
<td>Functional Security Requirements</td>
</tr>
</tbody>
</table>
### Implementation

<table>
<thead>
<tr>
<th>Sub-Activities</th>
<th>Description</th>
<th>Assigned Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparation of test environment(s)</td>
<td>The test environment should allow simulation of various faults as well as attacks</td>
<td>System Test Engineer, Security Manager, Safety Manager, Quality Manager</td>
</tr>
<tr>
<td>Validation of requirements</td>
<td>The implementation is validated against the functional safety/security requirements by using the test environment. Many safety and security tests can benefit from common test environment.</td>
<td>System Test Engineer, Security Manager, Safety Manager, Quality Manager</td>
</tr>
</tbody>
</table>

**Table 12 – Safety and Security Validation**

#### 3.5 Relationship to SESAMO Methods and Building Blocks

The following Table 13 illustrates the relationship between the activities of the generic process defined in section 3.4 and the safety and security analysis methods and building blocks elaborated in WP3 and WP2 respectively. The crosses in the intersection points of the table show the applicability of the methods (resp. building blocks) during the execution of the corresponding phases in the SESAMO process.

The various analysis methods can be grouped into the following rough categories:

*Integration of safety and security techniques for preliminary analysis*: this work pulls together or extends existing analysis methods (e.g., FMEA, FTA, Attack Trees) that are already part of standards for safety and for security so that the high-level design phase can identify hazards from the two viewpoints, and synergy and trade-offs in the mitigations and countermeasures.

Various probabilistic techniques for quantifying risk have been studied (extending the application of quantitative methods is one of the goals of WP3), with roles in analysing safety/security trade-offs and synergies at various levels of detail, to feed into the design phase as well as the validation phase.
Stochastic Modelling of Security and Safety at system level, in which a single stochastic model of the relationships between components and functions in a system describes the effects of both accidental failures and of attacks.

Quantitative modelling of Safety and Security for interdependent systems (e.g., embedded IT and the physical systems that it controls), which may need to combine heterogeneous quantitative models (e.g. stochastic models as above, models of attackers, deterministic models of how a certain failure of a system would propagate in others)

The above methods require parameter values for the models, and here security related parameters pose the greatest problems. To help with this, we have studied Quantitative security analysis with attacker strategy models that map knowledge or assumptions about attackers’ strategy, together with knowledge of the attacked system, to give probabilities of success for various attacks.

At a more detailed level, stochastic models have been built to decide Quantitative trade-offs in Communication Networks: especially considering the potential cost for communication reliability (with potential safety effects) of mechanisms introduced for security (viz. encryption increasing probability of communication being too late): here less uncertainty affects the parameters and these models can directly guide detailed design choice about protocols.

Last, all analyses need assumptions that are trusted with high confidence about a design: without them, both initial risk analysis and design validation would become unfeasibly complex. Formal methods give this kind of assurance, or flag to a designer when a design needs changes in order to ensure that assumptions required at system level are valid. In SESAMO, formal verification has been applied in two areas: Formal Security Analysis of Communication Protocols, checking properties of protocols for safety-oriented communication that have been augmented or for security properties; and Formal Analysis of Information Flow, allowing verification of how effectively the design of communication in a system prevents the propagation of localised failures or data corruption to cause damaging effects across the system.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Integration of safety and security techniques for preliminary analysis</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security-informed safety case</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stochastic Modelling of Security and Safety at system level</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Quantitative modelling of Safety and Security for interdependent systems</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Quantitative security analysis with attacker strategy models</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Quantitative trade-offs in Communication Networks</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Formal Security Analysis of communication Protocols</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Formal Analysis of information flow</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Work Package 2 - Building Blocks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Redundancy</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Partitioning</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitoring</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encryption</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

Table 13 – Relationship to methods and building blocks
4 MODEL-BASED PERSPECTIVE

Given its title of Security and Safety Modelling, it is not surprising that a model-based perspective is central to the SESAMO approach to an integrated design and evaluation methodology. One of the primary reasons for a model-based perspective is its capability to enable tool support for the methodology. Well-formed metamodels associated with the various steps of the methodology make it possible for tools implementing those metamodels to support the various phases of the lifecycle, including such SESAMO-specific artefacts and activities as the building blocks, trade-off analyses, and even certain aspects of safety / security case management.

Closely related to the general topic of tool support is the possibility of tool integration offered by the model-based perspective. Even if tools are implementing different metamodels (e.g. because of heterogeneous approaches to the modelling process itself), there is much more chance of at least partial integration among them – for example, through appropriately defined model-to-model (M2M) transformations. There is at least one specific example of such integration in SESAMO WP5, where two tools (medini analyze and CHESS), based upon non-identical modelling profiles, have undergone successful partial integration based upon M2M transformations.

A final reason for a model-based perspective on security and safety modelling is a general increase in precision: when a formal or at least semi-formal model is created, there is much less room for ambiguity, regardless of the level of tool-support that is available. This has proven to be useful in cases of re-certification or re-accreditation, where a model can be used to regenerate certification / accreditation documentation more reliably, even when it has to be verified by human eyes in the final analysis. (This is the case, for example, in Model-Based Security Engineering, where it has been demonstrated that even with the need for human intervention – mostly due to restrictions imposed by current standards – the re-accreditation process is streamlined and shortened through semi-automatic regeneration of documentation through models.)

4.1 CONTRIBUTIONS TO THE SESAMO MODEL-BASED PERSPECTIVE FROM CHESS

The generic SESAMO approach to joint security and safety modelling presents several challenges to the model-based engineering community:

- Safety and security are both non-functional properties. Traditional modelling approaches have been very good at capturing functional properties, but much less attention has been paid to non-functional properties – not just safety and security, but any non-functional properties such as performance and real-time characteristics (which also greatly affect safety and security, as the work in WP2 and WP3 of SESAMO has highlighted).

- The building blocks of SESAMO WP2, and to a great extent even the analysis methods of WP3, are intended to be reusable modelling artefacts. But traditional modelling approaches have placed little emphasis on the reusability of parts of models – often the concept of a “component” does not even exist in a modelling approach. There is a single model for the entire system and nothing else. But SESAMO needs the concept of a reusable modelling “component” that can be composed with others not only to form the overall system, but sometimes even the overall set of activities (e.g. different forms of trade-off analysis).

- Although the SESAMO generic process is described as a “joint” process, upon close examination it is evident that it involves a sequence of modelling first from different points of
view, in which there is a \textit{separation of concerns} between safety and security aspects, followed by confrontation involving trade-off-analyses – after which the cycle of separated modelling concerns followed by analyses starts anew. Traditional modelling approaches generally provide little or no support for modelling from different points of view – there is a single model, made from a single point of view. (We are not speaking of “vertical” abstraction levels here, which are well-known, but rather “horizontal” slices of the model, which are much less commonly provided.)

All of the issues described above must be addressed in order for the model-based perspective of SESAMO to be adequate to accommodate the generic process. The SESAMO solution to these issues is provided through the results of the \textbf{CHESS} (Composition with Guarantees for High-integrity Embedded Software Components Assembly) ARTEMIS JU Call 2008 project, aimed at developing solutions to property-preserving component assembly in real-time and dependable embedded systems [22].

In particular, CHESS promoted the adoption of Component-based Development and Model Driven Engineering to support the development of High Integrity Systems. It was the combination of these two approaches that provided the capability for reusability needed for the definition of the SESAMO approach. For the specific safety and security oriented needs of SESAMO we built upon the extensions that were introduced to the CHESS methodology and tool for the \textbf{SafeCer} (Safety Certification of software-intensive systems with reusable components) ARTEMIS JU Calls 2010 and 2011 project [53].

The CHESS project developed a methodology and related support technology based on four distinct pillars:

1. \textbf{Separation of concerns} by design views enacted directly in the user modelling space;
2. A cross-domain \textbf{component model} and a coherent component based development methodology;
3. \textbf{Correctness-by-construction}, by adoption of a declarative approach to the specification, verification and implementation of non-functional concerns.
4. Strict \textbf{separation between the platform-independent modelling} performed by the user and the platform-specific modelling, which is derived by formally verified transformation engines as a correct-by-construction product.

The \textbf{separation of concerns} concept is a long-known best practice for the development of complex systems – and, as noted above, it is needed to support the separate safety and security modelling activities of SESAMO. The methodological and technical means by which CHESS provides support for separation of concerns is the popular, well-accepted modelling concept of \textbf{views}, most prominently acknowledged in modern modelling languages such as UML. CHESS associates a distinct view with a distinct concern pertinent to system modelling, so that a particular concern (e.g. timing aspects, safety, security) is provided with a working environment devoted entirely to and specialized for that concern. In fact, the developer is \textit{only} allowed to work on one concern at a time – with access to the modelling environments of other concerns limited to read-only permissions.

The view-oriented development approach [11] of CHESS has required advances in component representation. The CHESS \textbf{component model} extends traditional component models by explicitly separating functional from non-functional aspects. Components internals are modelled in a dedicated view, in which the designer describes only the functional aspects. A distinct design view permits
to annotate the component description with the declaration of the intended non-functional attributes. The declarative specification of those attributes is used for the automated generation of a container, which can be considered as a component wrapper that is responsible for the realization of the non-functional attributes declared in the design model. As a consequence, in the CHESS approach pure components encompass functional concerns only; in particular, tasking or time-related constructs are a container concern.

Within CHESS itself, certain aspects of safety were addressed, within the context of general dependability, but few aspects of security were addressed. In contrast, within the EVITA project, security properties were addressed as a specific objective. The results of EVITA, and in particular its Model-Based Security Requirement Engineering process [10], can be a useful input for addressing the security aspects that were not addressed in CHESS. Note, however, that the modelling of safety and security requirements / properties was done in a different way in EVITA (as SysML requirements stereotypes and in the Temporal Property Expression Language [14]). In SESAMO we have preferred the container / wrapper approach of CHESS for new development, partly because of its strong support for component oriented development – although some use cases (in particular, the railway use case) continue to explore the use of the EVITA modelling approach.

The binding between two components in a CHESS design model is used to generate a connector, which manages the communication between the components – which is actually a mediated communication between their containers. The use of connectors ensures that components and containers do not require any adaptation under different binding and deployment specifications.

As a result, separation of concerns is directly supported in the CHESS model both at specification and implementation level: the former, by way of design views to address distinct concerns; the latter with the careful allocation of different concerns to distinct software entities.

The CHESS perspective on modelling is being experimented in SESAMO for supporting the generic safety and security process because it seems to provide a good fit:

- It introduces the concept of components and reusability into the modelling process, providing a way to support the ideas of “building blocks” that are incorporated into architectures;

- In order to support the idea that the building blocks have analysable properties associated with safety and security, we have taken over the concept of non-functional property modelling within a container / connector concept in the CHESS modelling perspective;

- Finally, in order to support the idea of modelling separate safety and security activities and then confronting results, we have taken over the CHESS support for separation of concerns.

4.1.1 SafeCer extensions to CHESS for safety arguments

As discussed elsewhere in this document, support for certification and re-certification of safety and security related systems is a long-term goal of SESAMO. Given that, as discussed above, the results of CHESS already provided several elements of interest for supporting the SESAMO generic methodology, the results of another project seemed appropriate to take into consideration for their possible contribution to the generic methodology.
The main goal of the SafeCer project is progress toward composable safety certification of safety-relevant embedded systems. A primary objective is to provide support for system safety arguments based on arguments and properties of system components, as well as to provide support for generation of corresponding evidence in a similar compositional way. The adoption of a **contract based** approach is an important contribution of SafeCer to the CHESS methodology: system and components’ properties are specified into formal expressions structured in contracts, i.e. pairs of assumption and guarantee, where the assumption is a property that must be satisfied by the (system/component) environment, while the guarantee is a property that must be satisfied by the system/component (provided that the environment satisfies the assumption).

Within each conceptual level the model describes a **step-wise refinement** with the hierarchical decomposition of the system into components and of components into subcomponents. Alongside with this architectural refinement, also the components’ contracts are refined into a collection of contracts over subcomponents, implementing an enhanced version of the traditional V-model development process as depicted in Figure 15. The formal V&V of the contracts at each level generates the safety evidence.

![Figure 15 – Extended V-model in the CHESS / SafeCer development process](image)

These extensions to the CHESS methodology are compatible with the SESAMO adoption of the enhanced V-model for its own generic methodology, and with the efforts of the consortium to support the generation of a joint safety and security case. The challenge that is only being addressed now is to understand the nature of security contracts – and even more difficult, contracts involving both safety and security; this is a challenge for which there is little prior experience to draw upon.

As mentioned earlier, the CHESS / SafeCer approach adopted within SESAMO for modelling with components, containers, and contracts is roughly comparable to the approach adopted in EVITA for modelling properties formally within Temporal Property Expression Language Parametric Diagrams and then undergoing formal verification. Both are being experimented within SESAMO through the use of toolsets that support the modelling styles.

### 4.2 MODELLING FORMALISMS

A discussion of the model-based perspective leads inevitably to the question of modelling formalisms. Coherent with its approach of “casting a wide net” over available formalisms for
modelling both safety and security, SESAMO has identified more than one formalism, creating both synergies and challenges.

4.2.1 UML-based formalisms

One important family of modelling formalisms in SESAMO is based on the Unified Modelling Language. This has become the modelling language of choice for those tools that are built within the Eclipse Modelling Framework (RMF) facility. For the kind of real-time, embedded systems modelled within SESAMO, however, UML is not enough, and some form of profile extension is needed.

One important profile, shared by several tools in the SESAMO consortium, is the SysML profile. SysML [50] adapts UML to offer a variety of modelling elements and diagrammatic views to support system engineering modelling activities: it can be used for specifying, analysing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities. In other words, it goes beyond the more purely software-oriented features of UML to provide the full system modelling features needed in SESAMO.

The AVATAR environment produced by the EVITA project, and also being experimented in SESAMO, is a graphical modelling environment that is even more heavily based on SysML. As noted earlier, in contrast to the container / component approach of CHESS, AVATAR expresses safety and security properties directly in terms of SysML requirement diagrams. There, safety properties are further refined within SysML parametric diagrams and security properties are described within specific pragmas of SysML block diagrams (this is reminiscent of the process of decorating the containers of CHESS with the safety and security properties).

The EVITA / AVATAR approach of directly using SysML artefacts (e.g. parametric diagrams, block diagrams, etc.) is one way to overcome the deficiencies of SysML with respect to dependability modelling. The MARTE UML profile [49] is another approach. It adds more explicit capabilities such as support for specification, design, and verification/validation stages, the definition of non-functional properties, time and time related concepts and analysis frameworks. At least one tool in the consortium (the CHESS tool) adds MARTE capabilities to the modelling language, in order to enable the specification of safety and security related properties in the components.

4.2.2 Other modelling formalisms

Aside from the UML-based family of modelling formalisms, other SESAMO modelling formalisms are associated with specific toolsets. For example, The BDMP (Boolean logic Driven Markov Processes) formalism was adapted from the dependability area to the security domain in order to model attack scenarios, and is supported by the KB3 modelling software platform of consortium partner EDF to let security analysts build and analyze security-oriented BDMP models. Similarly, the ASCE tool of SESAMO partner Adelard uses its own formalism for modelling security informed safety cases. The SCYThER tool uses a specialized modelling language, the Security Protocol Description Language, which can characterize security protocols and verify security claims.

The IMACT tool of partner SysGO is another example of the use of a specialized modelling formalism to implement a model-based engineering approach in the avionics domain of SESAMO.
As illustrated in Figure 16, the IMACT tool implements a classic model-based engineering approach with a specialized modelling language capable of representing the elements of the Integrated Modular Architecture (IMA) – including the SESAMO partitioning building blocks. While creating the model, the user resolves ambiguities and makes trade-offs as appropriate on the basis of analyses run on the models (this is where the SESAMO analytic techniques are applicable). Code generation can follow, much as in the AUTOSAR approach in the automotive domain. For further information on the IMACT workflow and technical details see Appendix A.

All of these specialized modelling formalisms co-exist within the SESAMO modelling environment, and are likely to be in use simultaneously within modelling initiatives, supported by different parts of the SESAMO toolchain. The degree of interoperability between those different tools will be determined by the possibility of property and semantics preserving M2M transformations between the formalisms used by the tools. This has been possible for those in the UML family, despite the use of different profiles and despite their different commercial providers.

It has also been possible when modelling tools come from the same overall family. For example, modelling tools that come from the SysML based AVATAR family and are in use within the consortium (see ProVerif [18] and UPPAAL [11] within the railway use case) have been successfully provided with automatic transformations from the original AVATAR models (and TTOOL [12] incorporates most of this in a single open source UML toolkit). Of course, these are “point-solutions” that have been provided exclusively within the specific family of tools.

Over the next year of SESAMO it will be determined (through work in the use cases) to which extent model integration is also possible for other combinations of tools using modelling formalisms whose semantics are not as immediately harmonized. Some tools provide “hooks” for implementing transformations to other modelling formalisms. For example, the IMACT tool produces XML formatted output that could be an input to a suitable model conversion mechanism.
5 ASSURANCE PERSPECTIVE

In this chapter, we provide an assurance perspective on the SESAMO methodology. In particular, we explain how the security-informed safety case approach that has been developed in WP3 relates to the SESAMO methodology and the integrated design and evaluation process developed in WP4.

Safety and security are both concerned with risk management and the purpose of assurance is to provide confidence that the safety and security risks have been reduced to an acceptable level. We provide this assurance in the form of a structured argument supported by evidence from the outputs of the SESAMO process. This argument can be developed in parallel with the development of the system and used to inform the development process. Essentially, at each stage of the V-model process, we construct a security-informed safety case at an appropriate level of abstraction to show why we believe that the safety and security risks have been adequately addressed.

5.1 RISK-MANAGEMENT

Safety and security are both properties of the overall system rather than any given sub-component. The top-level requirement is to maintain the safety or security of the plant, aircraft or other system. Thus, the construction of such systems focuses on minimizing and controlling risks. This is achieved by adopting a risk management approach, which aims to reduce the safety or security risk associated with the system to a tolerable level.

Risk management starts with a process of risk analysis that identifies the potential hazards or threats to the safety or security of the system and then analyses the consequence of an accident or breach of security. It is then necessary to decide how to deal with each risk, which is a process known as risk treatment. Possible strategies include risk avoidance, risk reduction, risk sharing, risk transfer, or risk acceptance. Some of these strategies involve making changes to the design or operation of the system. In particular, risk avoidance attempts to remove the underlying cause of the risk (the hazard or threat) by changing some aspect of the system’s design or operation. Similarly, risk reduction involves introducing mitigations or controls to manage the risk. These choices are recorded as additional safety or security requirements for the system, but since the system has now been modified, it is necessary to repeat the whole process of risk analysis and risk treatment until all risks have been reduced to an acceptable level, including any new risks that were introduced as a result of modifying the system. Moreover, in order to ensure that the system remains safe or secure, it is necessary to continually monitor the effectiveness of the risk management strategy and repeat the risk analysis to identify changes in risks.

The safety case or security case for the overall system makes a claim about the effectiveness of the risk management process. Targets are set for the tolerable frequency and severity of safety or security violations, and the top-level case argues that the implemented safety and security features ensure that the frequency of such violations is within limits. There is also a requirement to show that the risk is as low as reasonably practicable (ALARP), in other words, that the cost of reducing the risk further is disproportionate to the benefit gained.

5.2 ASSURANCE AND EVIDENCE

Assurance is about gaining confidence. Depending on the context, this could be confidence in the system as a product, or confidence in the operation of the system in a given environment. In the context of safety and security, assurance is our confidence that the safety or security risks associated with the system have been reduced to a tolerable level.
Assurance requires evidence, which can take many forms and is generated during the product development or operation of the system. For example, evidence from the development of a system would include the usual engineering documentation such as requirements specification, analysis and design documents, test specifications, implementation artefacts such as source code, and outputs from any verification or validation activities. Evidence from the operation of the system might include configuration data, change management history, performance statistics, incident logs and reports.

The degree of confidence that can be obtained in the assurance of a system depends not only on the strength of the claim that the evidence supports (or rebuts) but also on the extent to which the evidence itself is trustworthy. Thus, there needs to be confidence in the process for gathering, recording and maintaining evidence. Moreover, there needs to be some scientific basis for the techniques that are used to analyze the evidence and the conclusions that are drawn from this analysis.

5.3 DECISION SUPPORT

A structured safety case provides a method of making an informed decision about the trade-offs between safety and security. The primary goal is for the system to be safe, so we will always err on the side of safety rather than security. However, a system that is not secure cannot be safe. Therefore, we will construct a security-informed safety case that makes explicit claims about the safety and the security of the system, supported by appropriate arguments and evidence. This will enable us to make decisions about trade-offs and conflicts between safety and security in a principled way, and provide a justification for accepting that the risks associated with the system are as low as reasonably practicable.

5.4 SECURITY-INFORMED SAFETY CASES

In SESAMO, we have been exploring the use of structured safety cases for communicating and building confidence in the safety and security properties of a system. Security considerations have a significant impact on various aspects of safety justification. It is necessary to make claims about security properties as well as safety properties, demonstrate compliance to both security and safety standards, and consider a broader set of potential threats and vulnerabilities.

In principle, all safety cases should be “security-informed” because a system that is not secure cannot be safe. However, many safety cases assume that the system is operated in a benign environment, and therefore ignore security threats. This might not be an unreasonable position to take, providing the safety case makes this assumption explicit, and addressing security separately can be justified. However, in general, security must be considered as part of the safety case and any assumptions about security of the environment must be properly documented as part of the safety case.

Our experience and previous research has shown that a significant portion of a security-informed safety case will need to address security explicitly. In some instances this will lead to substantial changes to the design, the implementation process and the justification. In order to address these additional security risks within a case, it is necessary to find a way of combining safety and security risk assessment.
5.5 SECURITY-INFORMED RISK ASSESSMENT

Our method for performing a security-informed risk assessment is based on our experience of using risk assessment techniques to analyze large-scale critical infrastructure systems that need to be both safe and secure. The process consists of eight iterative steps to perform the risk assessment, as shown in Table 14.

<table>
<thead>
<tr>
<th>Step</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 – Establish system context and scope of assessment</td>
<td>Describe the system to be assessed and its relationship with other systems and the environment. Identify the services provided by the system and the system assets. Agree the scope of and motivation for the assessment and identify the stakeholders and their communication needs. Identify any existing analyses, e.g. safety cases.</td>
</tr>
<tr>
<td>Step 2 – Identify potential threats</td>
<td>Define the threat sources and identify potential threat scenarios.</td>
</tr>
<tr>
<td>Step 3 – Refine and focus system models</td>
<td>Refine and focus system models in the light of the threat scenarios to ensure that they are at the right level of detail for an effective risk analysis.</td>
</tr>
<tr>
<td>Step 4 – Preliminary risk analysis</td>
<td>Undertake architecture-based risk analysis, identifying consequences and relevant vulnerabilities and causes together with any intrinsic mitigations and controls. Consider doubts and uncertainties, data and evidence needs.</td>
</tr>
<tr>
<td>Step 5 – Identify specific attack scenarios</td>
<td>Refine preliminary risk analysis to identify specific attack scenarios. Focus on large consequence events and differences with respect to existing system.</td>
</tr>
<tr>
<td>Step 6 – Focused risk analysis</td>
<td>Match threat sources to attack scenarios and prioritize possible consequences according to the level of risk. As with Step 5 the focus is on large consequence events and differences with respect to existing system.</td>
</tr>
<tr>
<td>Step 7 – Finalize risk assessment</td>
<td>Finalize risk assessment by reviewing implications and options arising from focused risk analysis. Review defence-in-depth and undertake sensitivity and uncertainty analysis. Consider whether design-basis threats are appropriate. Identify additional mitigations and controls.</td>
</tr>
<tr>
<td>Step 8 – Report results</td>
<td>Report the results of the risk assessment to stakeholders at the appropriate level of detail.</td>
</tr>
</tbody>
</table>

Table 14 – Risk assessment process

In parallel with this process, a security-informed safety case is developed progressively throughout the risk analysis process to synthesize risk claims, arguments and evidence. The details of how se-
curity risks are mapped onto claims are very dependent on the specific case. Also, the case can be developed and issued at different levels of detail, depending on the intended stakeholder audience.

5.6 SECURITY-INFORMED SAFETY CASE METHODOLOGY

Because many systems already have safety justifications with corresponding risk assessments, we have developed a method for enhancing the safety justification with security considerations to develop a security-informed safety case. We start by expressing the safety justification for a system in terms of a structured safety case, with explicit claims, arguments, and evidence. This enables us to identify any gaps or omissions in the safety case or implicit assumptions that need to be made explicit.

In terms of methodology, the steps are:

- Express safety case about system behaviour in terms of Claims-Arguments-Evidence
- Review how the claims might be impacted by security
- Review security controls to see if these can be used to provide an argument and evidence for satisfying the claim
- Review architecture and implementation impact of deploying controls and iterate the process

Table 15 provides a brief description of each of these steps.

<table>
<thead>
<tr>
<th>Step</th>
<th>Brief description</th>
</tr>
</thead>
</table>
| Step 1 – Develop structured safety case | • Express the safety justification as a structured safety case  
  • Make claims, arguments, and evidence explicit  
  • Identify any gaps or omissions  
  • Highlight any assumptions that need to clarified |
| Step 2 – Develop security-informed safety case | • Identify security context and specific attack scenarios  
  • Review impact of security on existing claims, arguments, evidence  
  • Develop additional claims and arguments, identify required evidence, and revise structured safety case as necessary |
| Step 3 – Apply appropriate security controls | • Identify security controls that can be used to satisfy security claims, arguments, evidence  
  • Review impact of deploying controls on system life cycle |

Table 15 – Security-informed safety case methodology

5.7 LAYERED APPROACH

The justification of a security-informed safety case can be complex, or at least complicated, as it combines the claims from adaptation, supply chain and deployment, implementation details and hazard and vulnerability analysis. As one role of the case is to communicate effectively, one needs to balance both the risk of abstracting away important detail and the risk of the important detail being lost in a sea of other details.
Abstraction is a key structuring mechanism and we have experimented with various levels of abstraction when adding and structuring claims about the system. We call these abstraction levels – the layers of assurance, because within each abstraction level the assurance is composed. The main layers of assurance that we have identified are as follows:

**M0 Policy and requirements** – the highest level where the system represents its requirements, safety and security policies;

**M1 Architectural level** – the intermediate level where we analyze the abstract system components combined according to the abstract architecture;

**M2 Implementation level** – the detailed level where we analyze the implementation of specific components and their integration within the specific architecture.

These abstraction levels can be related to the initial phases of the SESAMO design process.

This layered assurance approach is relatively neutral as far as engineering processes are concerned. One of its strengths is that it provides a link between four important and interrelated processes:

1) The SESAMO process model for integrated design and evaluation
2) The risk management process
3) The safety lifecycle
4) The security lifecycle

A more detailed description of these abstraction layers can be found in the relevant chapter of D3.3.
6 INDUSTRIAL APPLICATION

6.1 AUTOMOTIVE ADAPTATION

This chapter gives an outlook onto a combined process regarding safety as well as security aspects. It is based on the safety standard for the automotive domain ISO 26262 [55], which was derived from IEC 61508 (see section 3.1). UML activity diagrams are used as a formalization method.

6.1.1 Lifecycle of the automotive safety standard ISO 26262

The automotive domain safety lifecycle is defined in the standard ISO 26262 (see Figure 17).

Figure 17 – Automotive Domain Safety Lifecycle

The lifecycle depicted by ISO 26262 corresponds exactly to the essential phases of automobile production. It defines three levels: Starting with the Concept Phase it runs through a series of development phases until it describes the activities for production and operation after the Start Of Production (SOP). Therefore the well-known V-Model is used (see Figure 18).
Furthermore the ISO 26262 distinguishes between system level development and then below it, both hardware and software development.

**6.1.2 Approach for a safety-security combined process definition**

Figure 19 shows the V-Model of the ISO 26262 already enriched by security aspects as an UML activity diagram.
Figure 19 – Formalized V-Model for a safety-security combined process

It starts with the Concept Phase, which produces the Functional Safety and Security Concept. The System Design activity takes this as an input in order to generate the Technical Safety and Security Concept. This concept will directly influence the development and is also used later to integrate and test the implementation. The produced test results and the formerly defined requirements from the Functional Safety and Security Concept are the base for the safety and security validation activities.

In the following these activities and their inputs and outputs are described in more detail.

6.1.2.1 Concept Phase
The Concept Phase (see Figure 20) comprises the Item Definition, the Hazard and Risk Analysis and the Functional Safety concept. All the activities emphasized in red are security related activities and thereby form an extension to the life cycle defined by the ISO 26262.
The **Item Definition** should collect all information relevant to the safety and security analysis and design for the item:

- purpose and description,
- function(s) and relations between functions,
- requirements for each function,
- draft architecture/outline,
- additional nonfunctional constraints,
- borders or interfaces to other items/systems,
- legal requirements,

whereby an item is a system that implements a function at the vehicle level. An important aspect is the identification of some malfunctioning behaviour of the considered system, which is later considered during Hazard and Risk Analysis. Potential malfunctions can be identified during Item Def-
inition using HAZOP analysis [54]1. Analogously the Item Definition activity is also extended by a security related activity that identifies possible security violations.

As a result of the Item Definition a functional description of the item, a set of malfunctions and a set of security violations are defined.

**The hazard and risk analysis** consists of three fundamental steps. The situation analysis and hazard identification considers the potential malfunctioning behaviour of the item in combination with certain operational situations that could lead to a hazardous event. These events are classified in the next step, the hazard classification, in respect to their severity (S), their exposure (E) and their controllability in order to determine the automotive safety integrity level (ASIL). The third step is to derive the safety goals that have to be fulfilled in order to prevent the hazards.

Similar activities can be defined for the security related aspects. During a threat analysis the threats have to be described and considered within certain operational situations. These threats can then be estimated in respect to their criticality and a corresponding criticality level might be derived. In the end security goals should be defined in order to reduce the risk of an attack.

The safety and security goals are now the input to derive functional safety and security requirements. In this phase first interference analyses have to be undertaken in order to identify their impact on each other.

**6.1.2.2 System Design**

The System Design Phase (see Figure 21) comprises the activities to derive technical safety and security requirements out of the functional requirement and to define a corresponding architecture. Again all the activities emphasized in red are security related activities and thereby form an extension to the life cycle defined by the ISO 26262.

1 This varies from the definition of the Item Definition activity in ISO 26262, where the HAZOP analysis is considered to be a part of the Hazard and Risk Analysis. Nevertheless, ISO 26262 allows for the process to be tailored, so it is permissible to also use HAZOP here.
In the safety area, supporting methods to derive technical requirements and analyze the system architecture include qualitative and quantitative Fault Tree Analysis (FTA) as well as Failure Mode and Effects Analysis (FMEA).

With regard to the security aspect, methods like Attack Trees and Security FMEA help to evaluate the system design.

The found design solutions for safety and security have to undergo an interference and impact analysis. Thus it can be assured that some built-in security mechanisms do not cause a malfunctioning of the safety mechanisms and vice versa.

Figure 22 shows that the same activities executed on the system design level are repeated for a corresponding hardware (resp. software) design.
6.1.2.3 System Development

The input of the system development phase is the technical safety and security concept. It consists of the requirements and architecture that are suitable to realize the safety and security needs of the system. Typically, the system development consists of the hardware and software development. That means, according to architectural decisions, parts of the requirements will be realized by hardware and others by software. As the trend is towards the usage of standardized or off-the-shelf hardware, an increasing share of the overall requirements will be realized by software.

An important activity in the development phase for software is to select the appropriate platform. Standardized platforms like AUTOSAR gain more and more importance, due to the flexibility they offer and the high degree of re-use. Furthermore, such platforms already bring a number of safety and security mechanisms with them, which just need to be configured and used in a way that allows them to satisfy the safety and security requirements of the technical safety and security concept. What happens is a mapping of the building blocks that are part of the technical concept to the concrete platform specific mechanisms. Using again AUTOSAR as the platform, the building block “communication protection” for a certain signal would be mapped to the AUTOSAR E2E library mechanism which implements the building block as part of the AUTOSAR base software. Usually, a configuration activity has to be done, as the concrete parameters of the mechanisms have to be set depending on the technical concept. In the mentioned E2E example, an appropriate E2E profile has to be selected in this activity.

Figure 22 – Safety and security activities of the Hardware/Software Design Phase

The result of the System Design Phase is the Technical Safety and Security Concept which is implemented later during the Development Phase.
The final code consists then of the platform code, configured to the needs of the technical concept and the application specific code, which also has to be provided according to the technical concept. Typically, generated code replaces more and more the formerly hand written code. Generation is e.g. possible form design models – in case of the AUTOSAR example, it is possible to generate the code of the software components from Simulink models and put this together with configured AUTOSAR base software code to obtain the final deployable code (the runnable).

These activities described above are shown in the diagram below.

Figure 23 – Safety and security activities of the Hardware/Software Design Phase

In order to perform the above mentioned process part (the development phase) for the AUTOSAR platform in an efficient manner, the SESAMO consortium has proposed a new standard specification to AUTSOAR – the AUTOSAR “Safety Extensions”. This standard defined, how the technical safety concept of an AUTOSAR based system can be expressed as part of the AUTOSAR tem-
plates. Furthermore, AUTOSAR’s methodology has been extended by activities and tasks to make use of such templates.

The current documents containing the approach in AUTOSAR are: [1][2][3].

As AUTOSAR is extending its capabilities towards security by introducing security mechanisms, it is intended to apply the same approach also for the security part of the technical safety and security concept. Once this is achieved, the methodology described herein is fully supported by AUTOSAR as the most widely spread automotive software platform.

6.1.2.4 Validation and Verification

The implementation has to be tested and verified against the requirements outlined in the Technical Safety and Security Concept as well as in the Functional Safety and Security Concept.

6.2 AVIONICS ADAPTATION

Safety is a major driver of the aviation industry and there is a well-established family of safety standards covering the entire system life cycle for both hardware and software, as shown in Figure 24.

Figure 24 – Overview of Safety-Related Development Assurance in Aerospace

This guidance on safety is now being augmented with guidance on security. The proposal is to develop two separate strands for safety and security within the overall system lifecycle, which is very much in line with the SESAMO approach to interaction with standards committees outlined in Chapter 7 of this deliverable.

6.2.1 Safety life cycle

ARP4754 discusses the certification aspects of highly-integrated or complex systems installed on aircraft, taking into account the overall aircraft operating environment and functions. The term "highly-integrated" refers to systems that perform or contribute to multiple aircraft-level functions.
The term "complex" refers to systems whose safety cannot be shown solely by test and whose logic is difficult to comprehend without the aid of analytical tools. The guidance material was developed in the context of Federal Aviation Regulations (FAR) and Joint Airworthiness Requirements (JAR) Part 25.

**ARP 4761** describes guidelines and methods of performing the safety assessment for certification of civil aircraft. It is primarily associated with showing compliance with FAR/JAR 25.1309. ARP 4761 describes the safety assessment process, namely

- Functional Hazard Assessment (FHA)
- Preliminary System Safety Assessment (PSSA)
- System Safety Assessment (SSA)
- Verification Means Used for Aircraft Certification

and the associated safety assessment analysis methods.

- Fault Tree Analysis/Dependence Diagram/Markov Analysis (FTA/DD/MA)
- Failure Modes and Effects Analysis (FMEA)
- Failure Modes and Effects Summary (FMES)
- Common Cause Analysis (CCA), which its specific analysis methods
  - Zonal Safety Analysis (ZSA)
  - Particular Risks Analysis (PRA)
  - Common Mode Analysis (CMA).

Several of these analysis methods are directly addressed either as SESAMO building blocks or as SESAMO analysis methods, while others such as Zonal Safety Analysis are typical for the avionics domain and not directly addressed at the SESAMO level. This is inevitable when domain-specific adaptations of the generic process are undertaken.

For the methods of compliance with the FAR and JAR 25 requirements for a new system design, five methodologies are generally adopted, some of which are described in more detail in ARP4754 and ARP4761:

1) Analysis including engineering analysis, stress analysis, system modelling and similarity modelling.
2) Failure Analysis including FMEA (Failure Mode and Effects Analysis), FTA (Fault Tree Analysis) and safety analysis (including Functional Hazard Assessment (FHA), (Preliminary) System Safety Assessment ((P)SSA), and Common Cause Analysis (CCA)).
3) Laboratory tests including component tests, qualification tests, system tests and through an integrated systems test rig.
4) Ground Tests – On aircraft ground tests.
5) Flight Tests.

In the SESAMO generic process, the primary emphasis is on techniques related to the first two items – the remaining items are highly domain-specific.

The SESAMO generic process provides the necessary conceptual framework to permit the inclusion of both software and hardware safety processes – which also exist in other domain-specific stand-
ards such as ISO26262 for the automotive domain. In the avionics context, RTCA ED-12C / DO-178C (Software Considerations in Airborne Systems and Equipment Certification) defines the software safety process required for safe implementation of software, whilst RTCA ED-80 / DO-254 (Design Assurance Guidance for Airborne Electronic Hardware) defines the hardware safety process required for safe implementation of hardware.

ED-12C provides recommendations for the production of software for airborne systems and equipment that performs their intended function with a level of confidence in safety that complies with airworthiness requirements. Compliance with the objectives of this standard is the primary means of obtaining approval of software used in civil aviation products. This use of an integrity/confidence levels approach is fully compatible with the corresponding generic SESAMO approach.

ED-80 is intended to help aircraft manufacturers and the suppliers of aircraft electronic systems assure that electronic airborne equipment safely performs its intended function. The document identifies design life cycle processes for hardware that includes line replaceable units, circuit board assemblies, application specific integrated circuits (ASICs), programmable logic devices, etc. It also characterizes the objective of the design life cycle processes and offers a means of complying with certification requirements. Most of these processes occur at a level that is more specific than the SESAMO generic process, but hooks are available especially with respect to making the evidence available for certification employing also the SESAMO security-informed safety case techniques.

The interaction between the safety and development process is shown in Figure 25, which is taken from ARP4754. Note that the classic V-model approach embraced by the SESAMO generic process is also used here, ensuring a fundamental compatibility.

**Figure 25 – Interaction between safety and development process (taken from ARP4754)**
Figure 26 shows the aircraft development processes in more detail.

<table>
<thead>
<tr>
<th>Safety Assessment Process</th>
<th>System Development Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aircraft Level Functional Hazard Assessment (FHA)</td>
<td>Aircraft Functions</td>
</tr>
<tr>
<td>Functional Interactions</td>
<td>Failure Conditions, Effects, Classification, Safety Requirements</td>
</tr>
<tr>
<td>System Level FHAs</td>
<td>System Functions</td>
</tr>
<tr>
<td>Failure Conditions &amp; Effects</td>
<td>Allocation of Aircraft Functions to Systems</td>
</tr>
<tr>
<td>Preliminary System Safety Assessments (PSSA)</td>
<td>Development of System Architecture</td>
</tr>
<tr>
<td>Failure Conditions, Effects, Classification, Safety Requirements</td>
<td>Architectural Requirements System</td>
</tr>
<tr>
<td>System Safety Assessments (SSA)</td>
<td>Item Requirements</td>
</tr>
<tr>
<td>Item Requirements, Safety Objectives, Analysis Required</td>
<td>Allocation of Requirements to Hardware &amp; Software</td>
</tr>
<tr>
<td>System Implementation</td>
<td>System Implementation</td>
</tr>
<tr>
<td>Certification</td>
<td>Physical System</td>
</tr>
</tbody>
</table>

Figure 26 – Aircraft development processes

6.2.2 Security life cycle

In order to address security, a number of new standards are being developed within this framework. In particular, RTCA/EUROCAE ED-202 / DO-326 (Airworthiness Security Process Specification) is intended to augment current guidance for Aircraft Certification to handle the Information Security Threat to Aircraft Safety. ED 202 adds data requirements and compliance objectives and is intended to be used in conjunction with other standards and regulation, particularly SAE ARP 4754A / ED-79A, DO-178C/ED-12, DO-254/ED-80.
An applicant for an airworthiness certificate complies with the guidance of this document by demonstrating that the process provides the necessary certification data and the process will satisfy the compliance objectives.

Figure 27 shows how security risk assessment activities fit into the development process V-model:

![V-model Diagram](image)

**Figure 27 – Security risk assessment activities in the development process V-model (taken from ED202A)**

The airworthiness security process (ASWP) is compatible with the overall approach of the SESAMO security-informed safety case, and provides interesting state-of-the-art input into the specific activities of assurance case preparation. ASWP comprises eight steps, which are shown in Figure 28. Step 1 is planning, Steps 2 to 4 are the risk management activities, Step 5 to 7 are the security protection development activities, and Step 8 communicates evidence.

1. **Step 1: Plan for security aspects of certification** describes the plan of activities to obtain certification. It is a joint activity of the applicant and the airworthiness authorities. It is an input to all further activities.


3. **Step 3: Security Risk Assessment** evaluates security risks and is equivalent to ISO 27005 [5].

4. **Decision Gate 4 “Are Security Risks acceptable?”** according to an airworthiness acceptability matrix, which defines an acceptable level of threat and severity of threat condition effects.
5. Step 5: Security Architecture and Measures. Security measures fulfil security requirements and are characterized through their security effectiveness requirements and security assurance actions.


7. Step 7: Security Verification and Validation. Validate security-related technical requirements and verify that systems meet the technical requirements and objectives and security effectiveness.

8. Step 8: Communication of evidences. The applicant communicates the evidences with the final Security Risk Assessment as means of compliance (Validation and Verification).

Figure 28 – Airworthiness security risk management framework (taken from ED202A)

RTCA/EUROCAE ED-203 / DO-YY3 (Airworthiness Security Methods and Considerations) is an emerging standard and describes guidelines, methods and tools used in performing an airworthiness security process. Practitioners are not bound by these practices as practices for airworthiness are still undergoing evolution and refinement as new features are deployed and the security threat itself evolves. Methods and considerations of ED-203 address the assessment of the acceptability of the airworthiness security risk and the design and verification of the airworthiness security attributes as related to system safety and airworthiness, specifically, the security scope, security risk assessment, security effectiveness and assurance, and security protection.
6.2.3 Integration between security and safety risk assessment

ED-202 combines the security and safety risk assessment, and as such, it provides an ideal domain-specific testbed for the approach to combined security and safety risk assessment in the SESAMO generic process.

According to ED-202: “The Security Risk Assessment related activities are a set of activities which interact with the safety assessment process to manage the added environment risk to aircraft when it is exposed to the threat of unauthorized interaction. It is analogous to a Particular Risk Analysis (PRA) in its interactions with ED-79A / ARP 4754A and ED-135/ARP 4761.”

And in more detail: “Security Risk Assessment is a set of Activities of the same nature as safety assessment activities. So according to the considered level, the aircraft and system level security risk assessment activities are related as follows:

- Preliminary Aircraft Security Risk Assessment (PASRA) is related to Aircraft Functional Hazard Assessment (AFHA) and Preliminary Aircraft Safety Assessment (PASA).
- Preliminary System Security Risk Assessment (PSSRA) is related to System Functional Hazard Assessment (SFHA) and Preliminary System Safety Assessment (PSSA).
- System Security Risk Assessment (SSRA) is related to System Safety Assessment (SSA).
- Aircraft Security Risk Assessment (ASRA) is related to Aircraft Safety Assessment (ASA).”

Figure 29 illustrates Security Risk Assessment related activities on different iterations and on different levels of the development (aircraft/system). Note the feedback and interaction between the corresponding activities, which can successfully be related to the concept of “points of interaction” being adopted in SESAMO (see discussion in Chapter 7).
Figure 29 – Airworthiness Security and Safety Assessment Process in Aerospace (taken from ED202A)

ED-202 identifies the relationships between security and safety at each level (cf. the SESAMO “points of interaction” discussed in Chapter 7). For example:

- **“Relationship between Threat Condition Identification/Evaluation and FHA”**

  Threat condition identification is the initial set of actions of Security Risk Assessment to determine the threat conditions. It is followed by actions to determine the threat scenarios and security risk.
Aircraft threat condition identification is linked to Aircraft Functional Hazard Assessment (AFHA), while the system threat condition identification is linked to the associated System Functional Hazard Assessments (SFHA).

Failure conditions are initial inputs for any threat condition identification and are provided by aircraft and system level Functional Hazard Analyses (FHAs) respectively. Those failure conditions are used for the identification of threat scenarios and associated threats causing them. In the second phase threats identified in the first phase are used to identify potential additional threat scenarios resulting in supplemental threat conditions causing safety effects. The severity of threat conditions is evaluated in the same manner as by Functional Hazard Analyses (FHAs). As such, any threat condition having an identical safety effect as a previously identified failure condition shares its severity.

- **Relationship between Preliminary Security Risk Assessment, Preliminary Aircraft Safety Assessment / Preliminary System Safety Assessments (PASA/PSSA) and Aircraft/System Architecture**

  Preliminary Security Risk Assessment may produce requirements for security measures which are provided to the process for Security Development.

  Architecture modifications (for example, adding of security measures) are to be analyzed by both Safety Assessment and Security Risk Assessment.


  Aircraft/System Security Risk Assessment and ASA/SSAs are performed on the same implemented architecture. The results of Security Risk Assessment are included in the Plan for Security Aspects of Certification Summary (Plan for Security Aspects of Certification (PSecAC) Summary).”

The avionics domain is one of the few domains that have begun standardisation of integrated lifecycle activities of safety and security. These activities provide a perfect domain-specific validation of the Hazard and Risk Analysis step of the SESAMO generic process, providing input to further iterations of the definition of the process to ensure maximum cross-domain compatibility.

### 6.2.4 Integration of security development activities in the development process

According to ED202A: “As far as Security Development activities are concerned, the concerned security activities are integrated into the aerospace development process as indicated in Figure 29. Aircraft Security Architecture and Measures activity is integrated into the Allocation of Aircraft Functions to Systems activity. The Aircraft Security Operator Guidance activity and Aircraft Security Verification activity are integrated into the Aircraft Level Integration & Verification activity. The System Security Architecture and Measures activities are integrated into the Development of System Architecture. The implementation of system security measures is integrated into the System Implementation like any other function implementation. The System Security Integrator Guidance activity and System Security Verification activity are integrated into the System Level Integration & Verification activity.”
Likewise, these initial domain-specific integrated safety and security related development activity
descriptions provide valuable input for further refinements of the generic SESAMO process, since
little is available in other domain standards to date to guide the process definition.

6.3 RAILWAY ADAPTATION

This section describes the way in which the lifecycle and approach to safe systems according to EN
50126 [62] for railway applications can be adapted to the SESAMO generic process.

As a member of the IEC 61508 “family” of standards, the railway family of standards demonstrates
a high degree of compatibility with the generic SESAMO process. The 50126 standard prescribes
steps in the lifecycle for the specification and demonstration of Reliability, Availability, Maintain-
bility and Safety (RAMS). In addition, there are the related standards EN 50129 [63] for safety-
related electronic systems for signalling, and EN 50128 [64] on software for railway control and
protection systems.

These railway standards do not, however, provide direct requirements for security, with the excep-
tion of the EN 50159 standard [65], which mentions security only in the context of communication
in transmission systems. It also provides links to standards for information technology – security
techniques. A more detailed discussion of the treatment of security in 50159 is provided in a later
section.

According to EN 50126, all factors influencing the system, including the external environment
should be considered. These factors are derived using cause/effect relations in a diagrammatic ap-
proach. As mentioned above, EN 50126 does not specify specific requirements for ensuring system
security; however, **security requirements can be considered in the context of external factors**
(hostile environment). This provides an opening for introducing joint safety and security activities.

Common safety methods are described in COMMISSION REGULATION (EC) No 352/2009 of 24
April 2009 on the adoption of a common safety method on risk evaluation and assessment [68] as
referred to in Article 6(3)(a) of Directive 2004/49/EC of the European Parliament and of the Coun-
cil [69].

6.3.1 The lifecycle according to EN 50126

Figure 30 shows the sequence of phases in the system lifecycle according to EN 50126. The V-
model includes a number of phases from concept of system to operation, maintenance, performance
monitoring and decommissioning.
Figure 30 – The V-Model according to EN 50126 [70]

Table 16 describes the individual lifecycle phases. Phase-related general tasks are common steps for fulfilling the objectives of the phase. Phase related safety tasks are focused on steps for fulfilment of safety objectives. In the final column, the mapping to the generic SESAMO safety and security related process is provided, together with a discussion of specific issues related to the nature of the railway-oriented process.

<table>
<thead>
<tr>
<th>Lifecycle phase</th>
<th>Phase related general tasks</th>
<th>Phase related safety tasks</th>
<th>Mapping to SESAMO process</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Define project concept.</td>
<td>Consider project safety implication.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Undertake financial analysis.</td>
<td>Review safety goals and policy.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Undertake feasibility studies.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Establish management arrangement.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. System definition and operational context</td>
<td>Establish system mission profile.</td>
<td>Evaluate past experience data.</td>
<td>Still part of Concept of Operations phase, in which scope exploration is carried out, and in particular, the relationship with existing systems and infrastructure is carried out.</td>
</tr>
<tr>
<td></td>
<td>Prepare system description.</td>
<td>Establish overall safety plan.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Identify operation and maintenance strategies.</td>
<td>Identify influence of existing infrastructure.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Identify operating condition.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Identify maintenance condition.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Identify influence of existing infrastructure.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Risk analysis and evaluation</td>
<td>Undertake project related risk analysis.</td>
<td>Determine the risk acceptance principles and criteria.</td>
<td>Hazard and Risk Analysis activity.</td>
</tr>
<tr>
<td></td>
<td>Perform system risk analysis.</td>
<td>Perform risk evaluation.</td>
<td>This is where Threat Analysis may be inserted to accompany the phase related safety tasks. See for example the discussion of threat analysis in EN 50129. Whereas the safety tasks contribute to determining SIL (see later discussion of SIL), a security criticality could also be determined here.</td>
</tr>
<tr>
<td></td>
<td>Set-up hazard log.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lifecycle phase</td>
<td>Phase related general tasks</td>
<td>Phase related safety tasks</td>
<td>Mapping to SESAMO process</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>5. Architecture and apportionment of systems requirements</td>
<td>Apportion system requirements: - Specify sub-system and component requirements. - Define sub-system and component acceptance criteria.</td>
<td>Apportion system safety goals and requirements: - Specify sub-system and component safety requirements. - Define sub-system and component safety acceptance criteria.</td>
<td>Still part of the Derive Functional Requirements activity of the generic SESAMO process. This corresponds to the activity of allocating the requirements to the preliminary architecture, but also to possible external measures.</td>
</tr>
<tr>
<td>6. Design and implementation</td>
<td>Perform design and development. Perform design analysis and testing. Perform design verification. Perform implementation and validation. Perform design of logistic support.</td>
<td>Implement safety plan by review, analysis, testing and data assessment addressing: - Hazard log. - Hazard analysis. Justify safety related design decision. Prepare generic product safety case. Prepare generic application safety case. Assess generic safety case.</td>
<td>This is the System Design phase of the generic SESAMO process. The safety and security architecture is developed. The security informed safety case is initiated. The safety related tasks in EN 50126 are augmented with security related tasks, whereby both safety and security related building blocks are applied. See following discussions of both safety measures and security measures in EN 50126 and EN 50129, demonstrating the compatibility with the generic process.</td>
</tr>
<tr>
<td>7. Manufacture</td>
<td>Perform production planning. Manufacture. Manufacture and test sub-assembly of components. Prepare documentation. Establish training.</td>
<td>Implement safety plan by review, analysis, testing and data assessment. Use hazard log.</td>
<td>This corresponds loosely to the Development phase of the generic process. Here the platform-specific components are implemented. Both safety and security hazard / threat logs are used. Building blocks are mapped to specific realizations in elements, with trade-offs resolved. This may include configuration data (common in railway applications).</td>
</tr>
<tr>
<td>8. Integration</td>
<td>Assemble system. Install system.</td>
<td>Establish installation programme. Derive safety-related application conditions.</td>
<td>This corresponds loosely to the Integration and Testing activities of the generic process. Note, however, that the EN 50129 activity Install System has a different mapping than in a standard such as the ISO 26262 automotive standard, which is aimed at mass production. No system is installed in such contexts. However, the generic process can accommodate both types of lifecycle.</td>
</tr>
<tr>
<td>9. System validation</td>
<td>Commission. Perform probationary operation period. Undertake training.</td>
<td>Implement commissioning programme. Update validation plan. Prepare specific application safety case.</td>
<td>This corresponds to the Safety and Security Validation activity of the generic process. In this instantiation of the generic process, there is a probationary operation period of the railway system. In mass production contexts, a different instantiation will be made such as captured fleets or long-term tests.</td>
</tr>
</tbody>
</table>
### Lifecycle phase Phase related general tasks Phase related safety tasks Mapping to SESAMO process

10. **System acceptance**
   - Undertake acceptance procedures, based on acceptance criteria.
   - Compile evidence for acceptance.
   - Entry into service.
   - Continue probationary operation period.
   - Assess specific application safety case.
   - This corresponds to the Independent Assessment activity of the generic process, where the role of the independent assessor is fundamental.
   - The security informed safety case is evaluated in this step.
   - A positive result is release for entry into service in the railway case. In a mass production context, it would be a Release for Production.

11. **Operation, maintenance and performance**
   - Long-term system operation, based on acceptance criteria.
   - Perform ongoing maintenance.
   - Undertake ongoing training.
   - Collect operational performance statistics.
   - Acquire, analyze and evaluate data.
   - Undertake ongoing safety centered maintenance.
   - Perform ongoing safety performance monitoring and hazard log maintenance.
   - Collect, analyze, evaluate and use performance and safety statistic.
   - The generic process does not go into detail for the operations and maintenance phases, other than to add security related considerations for these phases.

12. **Decommissioning**
   - Plan decommissioning and disposal.
   - Undertake decommissioning.
   - Undertake disposal.
   - Establish safety plan.
   - Perform hazard analysis and risk assessment.
   - Implement safety plan.
   - Decommissioning is not treated yet in detail in the generic process other than to note that there could be security related considerations in decommissioning (e.g. not disposing of sensitive data carriers in areas where malicious use could be made of it).

<table>
<thead>
<tr>
<th>Lifecycle phase</th>
<th>Phase related general tasks</th>
<th>Phase related safety tasks</th>
<th>Mapping to SESAMO process</th>
</tr>
</thead>
<tbody>
<tr>
<td>10. System acceptance</td>
<td>Undertake acceptance procedures, based on acceptance criteria. Compile evidence for acceptance. Entry into service. Continue probationary operation period.</td>
<td>Assess specific application safety case.</td>
<td>This corresponds to the Independent Assessment activity of the generic process, where the role of the independent assessor is fundamental. The security informed safety case is evaluated in this step. A positive result is release for entry into service in the railway case. In a mass production context, it would be a Release for Production.</td>
</tr>
<tr>
<td>12. Decommissioning</td>
<td>Plan decommissioning and disposal. Undertake decommissioning. Undertake disposal.</td>
<td>Establish safety plan. Perform hazard analysis and risk assessment. Implement safety plan.</td>
<td>Decommissioning is not treated yet in detail in the generic process other than to note that there could be security related considerations in decommissioning (e.g. not disposing of sensitive data carriers in areas where malicious use could be made of it).</td>
</tr>
</tbody>
</table>

#### Table 16 – Phases of EN 50126 lifecycle and mapping to SESAMO generic process

6.3.2 **Overall approach to risk management in EN 50126**

In this section it is examined whether the approach to risk management in the railway family of standards is compatible to the generic risk management approach of SESAMO.

The Hourglass Model (Figure 31) provides an overview of the major safety-related activities that are needed to ensure an acceptable safety level for a technical system, including the corresponding responsibility areas.

![Figure 31 – Hourglass Model [71]](image-url)
Note: In the figure, RAP = Risk Acceptance Principles (ALARP, GAMAB, MEM, etc.); ERE = explicit risk estimation.

Thus, it is apparent that the EN 50126 approach embraces the basic principles of overall risk management that are espoused by the SESAMO generic approach, and thus can indeed be considered to be compatible.

Continuing with the examination of the overall railway approach to risk management, risk analysis as defined in EN 50126 consists of these steps:

1. **Hazard identification** – recommended methods (e.g. FMECA, HAZOP) are the same as in other domains, only the list of typical scenarios is railway-specific.

2. **Hazard classification** - risks deriving from each identified hazard are categorised by a frequency and a severity. Controllability is not assumed as an independent parameter; rather, the standard uses a railway-specific term *barriers* for reducing frequency or severity.

3. **Consequence analysis** – includes a description of the relationship between causes and consequences of an event, recommended methods are the same as for safety analysis in other domains (e.g. ETA, FTA, FMEA). This is not railway specific.

4. **Risk evaluation and acceptance** – this is railway specific. The standard describes the following three methods:
   - The application of code of practice;
   - A comparison with similar systems;
   - An explicit risk estimation.

The above steps are in accordance with COMMISSION REGULATION (EC) No 352/2009) [7]. There is no fundamental conflict with the hazard analysis and risk assessment activities defined in the SESAMO generic process.

Likewise, there is no fundamental obstacle to flanking this hazard analysis and risk assessment activity with a companion threat analysis activity in an augmented railway development process. As mentioned several times in this section, a partial definition of such activities has already been started in the member standard EN 50129.

Figure 32 shows the approach to the risk management process and independent assessment according to this regulation [68]. It is entirely compatible with the overall generic risk management process defined in SESAMO, although with specific instantiations (e.g. Codes of Practice) that are particular to the railway field.
Figure 32 – Risk management process and independent assessment [68]
6.3.3 Functional safety integrity for E/E/EP

Continuing the examination of risk management / assessment in the railway standards, this section describes deriving functional safety requirements according to EN 50126. The current new draft of the standard gives an improved explanation of deriving THR (Tolerable Hazard Rate) for each sub-system. Here the difference is also described between THR and TFFR (Tolerable Functional Failure Rate), which is intended to replace THR for subsystems in the new version of the standard.

During the “apportionment of system requirements” phase, detailed functional and system requirements are derived from the Overall Functional Requirements. THR is derived through discussion, investigation and agreement between the railway duty holder and the supplier. The railway duty holder can also directly assign the THR to the technical hazards as a target to be reached by the supplier on the basis of the above stipulated risk analysis methods and principles. The approach here described is a top-down one (from tolerable accident safety targets to THR). Bottom-up method can be followed, starting from THR and verifying that the tolerable accident safety targets are observed. Figure 33 displays the two main aspects of hazard control.

![Figure 33 – Apportionment of functional safety requirements](image)

1. Initially, the quantitative integrity requirements (i.e., the Tolerable Hazard Rate, THR) for each hazard are apportioned to functions, by defining:
   - The safe state and its related fault negation time (permitted time to reach a safe state). Being in this safe state, the system is not allowed to fall back into a dangerous state if an additional failure occurs;
   - The Tolerable Functional Failure Rate (TFFR). This target may be specified at the level of the technical system, or it may be specified at the level of a sub-function of the system as it will always be specified at the lowest level of independence of the system function.

TFFR are rates \([h^{-1}]\). In the case that failure probabilities on demand are given, they must be transformed into appropriate continuous mode models.

It is important to note that the THR is a target measure with respect to both systematic and random failure integrity:
– It is accepted that only with respect to random failure integrity it will be possible to quantify and to verify that the TFFR target is fulfilled;
– Qualitative measures are also needed to enforce protection against random and systematic failures. This is covered by the measures derived from the safety integrity level.

2. In a refinement of the hazard control (after SIL allocation, when the safety-related function is apportioned in a number of sub-functions) the TFFR is further apportioned leading to failure rates for the subsystems/elements. The SIL remain unchanged at these lower levels of apportionment, as no further independence can be proven.

The concept of SIL is applied only for E/E/PE systems. The required SIL must be derived according to Table 17.

<table>
<thead>
<tr>
<th>SIL attribution</th>
<th>[TFFR] h⁻³</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>10⁻⁹ ≤ TFFR &lt; 10⁻⁸</td>
</tr>
<tr>
<td>3</td>
<td>10⁻⁸ ≤ TFFR &lt; 10⁻⁷</td>
</tr>
<tr>
<td>2</td>
<td>10⁻⁷ ≤ TFFR &lt; 10⁻⁶</td>
</tr>
<tr>
<td>1</td>
<td>10⁻⁶ ≤ TFFR &lt; 10⁻⁵</td>
</tr>
<tr>
<td>0</td>
<td>10⁻⁵ ≤ TFFR</td>
</tr>
</tbody>
</table>

Table 17 – SIL Table

Thus, Safety Integrity Levels in the railway standards are defined essentially in quantitative terms (like the mother standard IEC 61508), even though, as mentioned above, allowance is made for qualitative measures for dealing with systematic errors.

There is no concept of a Security Integrity Level yet in the railway standards, although first small steps have been taken with the classification scheme in EN 50129, as discussed later in this section.

In any case, the concept of integrity levels as expressed in the railway standards is fully compatible with its expression in the generic SESAMO process and activities.

6.3.4 An example of THR explicit risk estimation

A railway duty holder analyses a defined system from the hazard identification point of view, and during the analysis the hazard “Degraded ability to locate train” (e.g. faulty signalling equipment such as axle counters, train axle resistance too high for detection by track circuit) is reviewed. If a track circuit does not indicate when it is occupied it can cause this hazard. The railway duty holder calculates the hazard rate from statistics and by the expression

$$\lambda_{\text{hour per unit}} = \lambda_{\text{year}}/(\text{hours}_{\text{year}} \times \text{number of units})$$

whereby the hazard rate is estimated as $$\lambda_{\text{hour}} = 1.4\times 10^{-8}$$ failures per hour.

In next step a severity of the hazard is found out from the accident statistics with analysis whether a hazard of this type had led to any accidents and what consequences the accidents had. The railway duty holder creates a risk tolerability matrix e.g. as in Table 18.
Table 18 – The risk tolerability matrix [70]

A risk acceptance level can be determined for a given hazard rate and given severity in the matrix. For example if the determined severity of the hazard was III and estimated hazard rate 1.4E-08 failures per hour, the risk acceptance level is Tolerable (see red colour item). In this case the hazard rate 1.4E-08 failures per hour are Tolerable Hazard Rate (THR).

Note 1: in this example was used pessimistic assumption that each hazard leads to an accident.

Note 2: in this example the result of the risk analysis was Tolerable Hazard Rate of a hazard.

Although this realization of the process is specific to the railway industry, it is essentially an implementation of the generic ALARP principle and the general risk management principles described in this document, and no conflict is seen with the generic SESAMO process.

6.3.5 The railway “building blocks and analysis methods” [72]

The standard IEC 60300-3-1 (Application guide – Analysis techniques for dependability – Guide of methodology) [66] provides a comprehensive overview of safety/hazard analysis techniques in conjunction with their usability.

An overview of safety and risk analysis methods that can be used for railway application development is given in Table 19.

<table>
<thead>
<tr>
<th>Technique/Method Ref. to standard</th>
<th>Hazard identification</th>
<th>Qualitative analysis</th>
<th>Quantitative analysis</th>
<th>Safety demonstration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hazard and Operational Analysis (HAZOP) IEC 61882</td>
<td>YES.</td>
<td>Identification causes and consequences.</td>
<td>NO</td>
<td>Partially useful.</td>
</tr>
<tr>
<td>Rapid Ranking Analysis (RRA)</td>
<td>YES (partially)</td>
<td>Useful for preliminary hazard analysis and for identifying and ranking hazards for following detailed analysis.</td>
<td>NO</td>
<td>Possible for recording rationale for not performing detailed analysis of low ranking hazards.</td>
</tr>
<tr>
<td>Failure Mode, Effects and Critically Analysis (FMECA) EN 60812 [6]</td>
<td>YES</td>
<td>Identification consequences of failures. Useful for parallel structures in addition to ETA.</td>
<td>Calculation of failure rate and their severity.</td>
<td>Useful for single and parallel structures in addition to FTA for causal analysis.</td>
</tr>
</tbody>
</table>
### Table 19 – Overview of safety/risk analysis methods [71]

Thus, the railway standards are constructed along the same scheme as other standards in the IEC 61508 family in particular, as well as the SESAMO generic process: the methods listed above are essentially building blocks and analysis methods, and can be integrated and extended with the SESAMO building blocks and analysis methods, both concerning safety and security.

#### 6.3.6 Security approaches

Railway standards do not determine requirements for security, as mentioned earlier. An exception is the standard EN 50159, which defines security requirements only for communication in transmission systems. These security requirements have been extended by further requirements after analysis. The tasks of security should be integrated into all phases of life cycle like tasks of safety.

For example, EN 50159 provides a classification scheme for transmission systems. A transmission system is assigned a class according to defined criteria (Pr1, Pr2, Pr3). The standard defines threats and requirements for defences. Each classification of a transmission system has to consider concrete threats and their relation with appropriate defences. Categories and threats are described briefly below:

- **Category 1** – Closed transmission systems, where all essential properties of the system are under the control of the safety-related system designer, and a simplified set of safety requirements can be defined;

- **Category 2** – Open transmission systems where, although the transmission is not fully under the control of the safety-related system designer, the risk of malicious attack can be considered negligible;

- **Category 3** – Open transmission system where there is opportunity for malicious attack, and cryptographic defence measures are required.

Table 20 lists relations between threats and defences:
### Table 20 – Threats/Defences matrix [65]

<table>
<thead>
<tr>
<th>Threats</th>
<th>Repetition</th>
<th>Deletion</th>
<th>Insertion</th>
<th>Re-sequence</th>
<th>Corruption</th>
<th>Delay</th>
<th>Masquerade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Time stamp</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timeout</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source and</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>destination</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>identifiers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feed-back</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>message</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identification</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>procedure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Safety code</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Cryptographic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>techniques</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Only applicable for source identifier. Will only detect insertion from invalid source. If unique identifiers cannot be determined because of unknown users, cryptographic techniques shall be used.
- Application dependent.

Other security requirements of subsystems and applications are outside the scope of EN 50159. However, new methods for the integration of security aspects may be possible to implement according to other standards, for example ISO 15408 – Common criteria [12] and NIST SP 800-53 for Recommended Security Controls [74].

Two observations are appropriate at this point:

- The system of categories is an embryonic scheme for arriving at security integrity levels, not fully developed by any means, but the beginnings of an attempt to include the concept in a standard;
- The threats and defences matrix corresponds to a table of security related building blocks.

Thus, the direction in which the new suite of railway standards is headed is very compatible with the approach captured by the generic SESAMO process. It provides an excellent example of an instantiation of the generic process in a domain that is not oriented at mass production (like, for example, the automotive and medical domains), and therefore provides an opportunity to study issues with the process that may need adjusting in this type of context.

### 6.4 Industrial Process Control Adaptation

This chapter gives a first proposal for a safety and security enabled process for the industrial domain. The underlying safety lifecycle in industry is the IEC 61508 which is interwoven with the generic process proposal defined within SESAMO. This process was created by starting with an analysis of the generic SESAMO methodology including all its activities and finding similar process steps in the industrial safety lifecycle, which was shown in section 3.2. These similarities are then matched against each other and all extensions (in the figures these areas are marked with the keyword “extension”) to the SESAMO generic process are commented.

The generic process includes the phases:

- Scope Exploration
- Hazard and Risk Analysis
- Requirements Derivation
- System Design
6.4.1 Scope Exploration

Figure 34 shows the adaption of the two phases from the industrial domain process to the generic Scope Exploration phase. The respective phases Concept Phase and Scope Definition are in accordance to the IEC 61508 and brought into relation with the respective activities from the generic process:

- Concept Phase: Main objective is to get a better understanding of the system, which includes the identification of system functions (Identify Functions), and to gain preliminary hazard identifications.

- Overall Scope Definition: This phase defines the system boundaries, physical equipment, external events, equipment and systems that are associated with hazards. All that latter information will serve as a base for later in the process following Hazard Analysis and Risk Assessment. Furthermore, all possible malfunctions are identified (Identify Malfunctions) with information pro-
vided from the preliminary hazard identification in the Concept Phase. These malfunctions are found by using systematic approaches, such as a HAZOP (Hazard and Operability) study. The original industrial process is extended (“extension” in Figure 34) with security weaknesses that are identified by activities taking place during the Identify Security Violations activity.

- Results of the Scope Exploration phase is a system description including system functions, identified malfunctions and identified security violations.

6.4.2 Hazard and Risk Analysis

![Figure 35 – Hazard Analysis and Risk Assessment related to the Industrial Domain](image)

The adapted Hazard Analysis and Risk Assessment (HARA) phase (see Figure 35) has the aim to determine hazards and risks for the EUC (Equipment under Control), and with the extension of the generic process, security threats are considered as well. During the Hazard and Risk Analysis, the hazards are identified for the EUC and the ECS (Electronic Control System) and integrated into a risk matrix. SIL levels and safety functions (top-level safety requirements) are identified (activity Derive Safety Goals) - this is aided by FMEA (Failure Mode and Effect Analysis). Then the SRS (Safety-Related System) is assessed using reliability modelling in order to check whether the intended target SILs are reasonable for the SRS or if further RRM (Risk Reduction Measures) are necessary in consideration of the ALARP (As Low As Reasonably Practicable) concept. The func-
tional description and previously identified malfunctions are used as a basis for the HARA. Outputs of these activities are top-level safety requirements (called safety goals within the generic process) together with hazard scenarios, risk and safety level classification matrix, risk reduction measurements, (additionally) identified safety-related functions and a proven reliability diagram for the SRS.

The original Hazard and Risk Analysis in the industrial domain process considered only safety aspects, while the adapted industrial process also considers security issues. Based on the functional system description and the former identified security violations, potential security threats to the system are identified. These threats are then rated by their criticality level and security goals are derived, this is aided by a security FMEA. With this knowledge, operational situations are evaluated to determine the interaction effects between safety and security properties, which might lead to the need of balancing safety and security requirements and efforts for the future design.

6.4.3 Requirements Derivation

![Figure 36 – Requirements Derivation related to the Industrial Domain](image)

The Requirements Derivation phase (Figure 36) includes several activities that are mapped to the industrial process including the phases Overall Safety Requirements and Overall Safety Requirements Allocation:

- Overall Safety Requirements: Based on the previously defined Safety Goals, a set of overall safety functions is created (activity Derive Functional Safety Requirements), which includes the
safety function requirements and the safety integrity requirements (a SIL level is assigned to each safety function). The extension of the original industrial process comprises Derive Functional Security Requirements based on the former defined Security Goals. Security and safety function requirements are then checked for interferences and they are balanced if necessary. Results are Functional Safety Requirements including safety function requirements (independent from implementation technology) and safety integrity requirements, as well as a table assigning SILs to safety functions. Further results are Functional Security Requirements and Preliminary Architecture, which serves also as a base for the following overall safety requirements allocation.

- Overall Safety Requirements Allocation: The functional safety requirements are mapped onto the designated SRS described in the preliminary architecture. As extension, the functional security requirements are mapped as well.

### 6.4.4 System Design

![System Design related to the Industrial Domain](image)

**Figure 37 – System Design related to the Industrial Domain**

The System Design phase (see Figure 37) is related to the Overall Safety Requirements Specification phase from the industrial safety process.

System Safety Requirements Specification: As a first step, the technical safety requirements are derived from the allocated safety requirements from the previous one. The Safety BBs are matched as far as possible to match the technical safety requirements. The fulfilment of all the necessary safety requirements can be a combination of a design with standard Building Blocks taken out of
the SESAMO building block library and customized safety blocks. With the definition of the appropriate interfaces for all safety functions (time, data) and operation modes, and the definition of all safety integrity requirements (SIL) for each safety function including required duty-cycles, lifetime, possible environmental conditions with constraints/limits, the technical safety requirement document is fit for further refinement in the following Software and Hardware Design phase in the generic SESAMO process.

Analogously, technical security requirements are derived, and security Building Blocks applied. With the system design at hand, an interference analysis between safety and security is executed, which might lead to further refinements and balancing of the system design concerning a reasonable security-safety trade-off.

The results are technical safety and security requirements as well as safe/secure system design document.

6.4.5 Hardware and Software Design

![Figure 38 – Hardware and Software Design related to the Industrial Domain](image)

The Realization phase of the original industrial domain safety process is split into two parallel requirement specification phases: Requirement Specification for Safety-Related System / Software Design (Design and Development for system and software from the original industrial safety process will follow in the Implementation phase of the generic SESAMO process).
Requirement Specification for Safety-Related System / Software Design: Based on the system design phase further requirements are derived for the respective hardware and software parts of the design. Hardware and software Building Blocks are assigned respectively in combination with an interference analysis resulting in a hardware architecture and a software architecture together with the SW/HW safety and security requirements. FME(C)A / FTA, and their security equivalents aid in the requirements elicitation. Resulting technical documents will be the foundation for the implementation of software and hardware.

6.4.6 Development

![Image of Development process diagram]

Figure 39 – Development related to the Industrial Domain

The Development phase relates to the Safety-Related System Realization phase of the origin industrial safety process. There are several activities in the generic SESAMO process, such as the choosing of the implementation platform, the mapping of Building Blocks onto the platform (which is definitely an extension to the industrial safety process), the derivation of configuration information, and the generation/implementation of code.

Results are the realized safety and security related systems, including safety and security related software, accompanied by conclusive documentations.
7 DISCUSSION

The definition of the SESAMO methodology is not taking place in a vacuum. Given the nature of the mission-critical systems to be developed with the methodology, an important context exists that cannot be ignored: the standards communities. The safety community in particular has established standards in a number of domains governing safety-related embedded systems development, including avionics, railway, and the automotive industry. A number of security related standards have also been developed.

Given the importance of both safety and security in mission-critical systems, it is natural to consider the inclusion of both within a single standard. However, to date there has been little cross-fertilization between the safety and security communities developing these standards [60]. Even the vocabulary used in the two communities is often confusing and contradictory [59]. Thus, incorporating security requirements into safety related standards (and vice versa) has been extremely challenging to date.

Nevertheless, an increasing amount of activity is becoming evident, both in the technical community (e.g. [57], [61]) and in the standardization committees, reflecting the increasing need within the embedded systems industry. An example in standardization is the latest version of IEC 61508, the domain-independent safety standard for electrical and electronic systems. In its latest edition, it introduces for the first time requirements related to security. Clause 1.2 k) of Part 1 of the standard “… requires malevolent and unauthorised actions to be considered during hazard and risk analysis …”

This opens the door to the integration of security requirements also into the domain specific safety standards for which it serves as the “mother” standard. This process has already begun: for example, Draft EN 50126-5:2012 states that “… the Safety Case shall demonstrate that […] misuse-based failures on external interfaces do not adversely impact on the safety integrity of the system.”

However, this “opening” is in at least one respect less a door than a Pandora’s Box. Once the step is taken of including both types of requirements in a standard, a much more difficult problem comes into the foreground: the process. Mature development processes for safety related embedded systems have existed for years, and lifecycle activities for security related development (e.g. [58]) have also been defined. But the integration of safety and security related development processes has been a controversial topic in the standards communities.

A case in point is the automotive safety standard ISO 26262, another of the domain-specific “children” of the generic IEC 61508 standard. In 2013, the ISO 26262 standardization committee (ISO TC22/SC3/WG16) began to debate the inclusion of security related considerations in the standard. During the debate, a strong position was presented that security and safety are very different areas, requiring different skills, and that the respective processes must remain separate (see also [56]). This position argued that, rather than including specific security requirements in the standard, interaction points between the two (separate) processes be identified, and requirements defined to establish appropriate communication channels. Another position presented preliminary ideas for what could happen at the interaction points, such as combining FMEA with security analysis (e.g. by adding failure modes to the FMEA representing security attacks).

This concept of establishing points of contact between parallel safety and security lifecycle activities is also more or less in line with the approach being taken within the EUROCAE/RTCA com-
munity, where on-going work is examining the addition of security related activities to the heavily safety-oriented ARP4754 system development process.

SESAMO has declared many times that its ultimate goal is a unified safety and security process. However, to paraphrase the old adage, “an unused process is a process developed in vain,” and a realistic approach to the standards communities is necessary. Therefore, the SESAMO approach is to align itself with the idea of well-defined interaction points as adopted by several standards communities to date. Figure 40 illustrates how SESAMO positions itself within this scheme.

![Figure 40 – Safety and security lifecycle activities](image)

In this scheme, two types of SESAMO intervention can be identified:

1. **Trade-off (or “analyses at the interaction points. These are particularly relevant for those lifecycle activities and mechanisms that tend to be purely safety or security related. For example, the use of cryptography for confidentiality is a security issue rather than a safety issue, although cryptography can also be used to achieve integrity and authenticity in the safety domain. The analysis methods developed in SESAMO can make it possible during communication at the interaction points to judge the effects of the activities in each parallel process on the other, and to provide appropriate decision support based on the results. Note that this tends to concern the building blocks that are architectural in nature.

2. **Joint lifecycle activities. This is a deeper intervention, where the lifecycle activities are actually combined, such as joint hazard and threat analysis, or joint FMEA and attack analysis. Note that this tends to concern the process building blocks rather than the architectural building blocks – but it is not always the case. For example, a “redundancy” building block (and its accompanying analysis methods) could be instrumental in a joint safe and secure architectural design activity.

An important point to observe is that the SESAMO approach assumes that, although the processes might be parallel, they are working on a single set of workproducts: the two types of intervention
described above are intended to support the development of this single set of workproducts with the appropriate set of safety and security attributes.

The approach of parallel processes with “weak” trade-off interactions and “strong” interactions for joint activities has the advantage of providing a smooth migration path for the standards communities, allowing them to start with the separate processes of today and gradually identify and implement architectural and process building blocks that promote an ever-closer integration of the processes, while continuously approaching the Holy Grail of a fully integrated SESAMO process.
8 SUMMARY AND CONCLUSIONS

For a system to be safe, it also has to be secure. Otherwise, a safety-critical system – one that can harm or injure people – could provide attackers with a potential mechanism for causing widespread damage or panic, and it is credible that such systems could become the target of malicious actions.

In principle, achieving interworking between safety and security should be straightforward. Both are sophisticated engineering cultures that emphasise the need for good process, the importance of risk analysis and the need for assurance and justification. However, in practice there are significant challenges to be overcome, as our experience with SESAMO has shown.

To illustrate the complexities of the problem and to demonstrate the progress we have made, we conclude this deliverable by summarizing some of the technical issues involved in combining safety and security assurance in a principled way.

8.1 CONCEPTS

The commonalities between safety and security are frequently obscured by the use of different concepts and terminologies. Indeed, there is considerable variation in terminology both within and between the safety and security communities. Thus, to achieve a shared understanding of the key concepts within each domain, there is a need to establish a lingua franca or even a common ontology.

The IFIP WG 10.4 dependability taxonomy [77] offers some hope for defining a consistent set of terms. In particular, it makes a clear distinction between cause and effect and highlights the need to be clear about system boundaries.

Broadly speaking, safety is concerned with protecting the environment from the system whereas security is concerned with protecting the system from the environment. Security and safety can both be viewed as kinds of dependability and use similar techniques to identify potential failure modes and assess their impact on the overall system. Thus, there is considerable overlap between safety and security methods, although the focus is different and in some cases safety and security requirements can be in conflict.

In particular, one of the major differences between safety and security is that a secure system needs to cope with evolving threats and changes to the environment through design and architectural measures as well as operational ones. It is important for the system to remain safe and secure despite such changes, in other words, to be resilient to change.

8.2 METHODOLOGY

Risk assessment is a fundamental step in safety and security analysis, but the underlying threat model is different. Thus, an important part of the SESAMO development process is a unified methodology for assessing the threats to the safety and security of a system.

This was accomplished by first starting with an overarching approach to risk management as laid out in ISO 27005, followed by an explicit adoption of the ALARP principle, thereby introducing a concept of (tolerable) levels of risk. Unifying the methodology around the concept of risk levels made it possible to introduce and adopt the well-accepted approach of prescribed techniques and measures to achieve the appropriate levels of risk (whether safety or security related). In turn, this provided the “hooks” needed to make the all-important connection to the building blocks and analy-
sis techniques that are at the core of the SESAMO initiative. Basing the overall process flow on the well-accepted V-model rendered the unified process amenable to mapping into several domains of interest.

8.3 MODEL-BASED DEVELOPMENT

Modelling formalisms are applied pervasively in order to facilitate tool support for the methodology (indeed, the steps of the generic process have been already modelled in at least one of the support tools and are foreseen for others). Another foreseen application of model-based development techniques is in the exchange of data and models between tools, since it has become quickly clear that full support for the methodology, spanning the full range of building blocks and analysis techniques provided by SESAMO, will only be possible with an extensive, probably heterogeneous tool set, giving rise to challenging integration issues. Finally, the very act of formalising the process has promising implications within the assurance arena, where re-assurance after modification can be facilitated through keeping the assurance in models. This is related to the discussion of assurance in the next section.

8.4 SECURITY-INFORMED SAFETY CASES

Security considerations can have a significant impact on a safety case. For example, there needs to be an impact analysis of the response to security threats and discovery of new vulnerabilities and reduction in the strength of protection mechanism. This suggests a greater emphasis on resilience of the design.

It is also necessary to consider the potential for attack during a safety incident and the opportunity this might provide for malicious activity. A fail-safe state may not be as safe as previously thought if the system is under attack and the assumption that any security attack on a control system could only, at worst, cause a fail-safe state to be reached is in general not true. Moreover, assumptions about the capabilities and state of society may change; for example, consider managing a safety incident during a major security incident.

Given the importance in security of open scrutiny of design and implementation (e.g. of crypto), it is worth considering whether security-informed safety cases should be disclosed in their entirety or in part. Within the safety community, the principle of independent assessment is well established, but the design details within a safety case are usually considered to be confidential and are not made public in their entirety.

The key question is whether publishing the detailed design and safety analysis for a system would make the system less or more safe and secure, or more precisely which aspects would it be beneficial to expose and which not.

8.5 STANDARDS

Safety standards already require “malevolent and unauthorized actions to be considered during hazard and risk analysis” [78], and there have been a number of domain-specific attempts to define a unified approach to safety and security assurance [79][80]. However, the standards framework for dealing with security-informed safety needs to be more explicitly designed than is currently the case. In particular, the relationship between generic and domain-specific safety and security standards needs to be clarified, and terminological and conceptual differences need to be resolved.
The standards framework should be based on explicit principles and use a consistent terminology. The standards groups should ensure they have available a suitable mix of both security and safety expertise.

Security standards are often based on security controls, a concept that embraces a wide range of different interventions covering process, product and organisation. In contrast, safety standards are typically based on an engineering life cycle model. One of the goals of the SESAMO methodology is to find a way of combining both approaches within a common framework.

Standards often use “levels” as a way of classifying systems, risks and controls. However, it is important to understand the assumptions that underpin these classification schemes and not to confuse different kinds of classification. In particular, risk levels, requirement levels, and assurance levels need to be carefully distinguished.

A particular concern is the problem of justifying requirements that specify the use of particular methods and tools to achieve a specific level. In order to support interworking between safety and security standards, there needs to be a better understanding of the rationale for such recommendations and the evidence base that supports them. The work that SESAMO has done on analysing the trade-offs between safety and security mechanisms is particularly relevant here.

8.6 CONCLUSIONS

With the successful elaboration and description of a generic process, its underlying principles and concepts, steps, and activities, much has been accomplished. But further challenges lie ahead as it is applied in the field within the context of the SESAMO use cases. Initial experience is promising. It has been possible to apply the generic process and associated methodology in the key SESAMO domains with a generally reasonable matching to existing processes, and tool support is proving possible for many of the building blocks and analysis techniques. However, consolidating the many individual applications of the SESAMO results into a coherent, practical, and integrated approach with associated tool support will be the major challenge to be faced moving forward.

References


[22] “D2.3 – Multi-concern Component Methodology (MCM) and Toolset” Version 1.1, 21 January 2011


R. Kissel (Ed.), Glossary of Key Information Security Terms, NIST IR 7298, Revision 1, February 2011.


[63] EN 50129 Railway application – Communication, signalling and processing systems – Safety-related electronic systems for signalling.

[64] EN 50128 Railway application – Communication, signalling and processing systems – Software for railway control and protection systems.

[65] EN 50159 Railway application – Communication, signalling and processing systems – Safety related communication in transmission systems.

[66] IEC 60300-3-1 Application guide – Analysis techniques for dependability – Guide of methodology

[67] EN 60812 – Analysis techniques for system reliability – Procedure for failure mode and effects analysis (FMEA).


[70] Draft prEN 50126-1 Railway applications - The Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS) - Part 1: Generic RAMS process

[71] Draft prEN 50126-2 Railway applications - The Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS) - Part 2: Systems approach to safety

[72] Sesamo project D3.1 UC

[73] ISO 15408 – Common criteria

[74] NIST SP 800-53 for Recommended Security Controls


A IMA

An IMA (Integrated Modular Avionics) system is a defined set of hardware components providing computational, communication, and interface capabilities for hosting applications realizing aircraft functions or parts of aircraft functions. The hardware components comprised by IMA systems are RDCs, Controllers, AFDX End Systems, IPMs (Integrated Processing Modules), Cabinets and AFDX switches.

A.1 IMACT OVERVIEW

The IMACT (Integrated Modular Avionics Configuration Tool) is intended to model IMA systems and to generate configurations for the involved hardware components based on input specifications (Usage Domain Definitions, Hosted Function Requirements). System Function Requirements are defined as a result of the IMA platform used to realize the Hosted Function Requirements.

Based on input specifications the topology and communication structure of the IMA system is modeled. The user is able to perform analysis regarding consistency, determinism and reliability of the modeled IMA system against the input specifications. Finally, configuration files for the hardware components of the IMA system as well as internal interface descriptions are generated by the IMACT.
IMACT requires at least the hosted function requirements and usage domain definitions of an IMA system as input.

Hosted Function Requirements specify the resource and communication needs to realize a hosted function (required IMA interfaces, computational capabilities). Also hosted function requirements specify hosted application partition’s resource requirements and redundancy requirements. The hosted function requirements specify the ATA chapter they provide functionality for. A hosted function is the part of an aircraft function which uses the shared resources of the IMA system. Each aircraft function can be realized by one or multiple hosted functions, is derived from an ATA chapter, and refers to specific functionality of an aircraft. Aircraft functionalities not using shared resources of the IMA system are not visible to IMACT.
Usage Domain Definitions are the technical description of components used within IMA systems. They describe the device capabilities (e.g. interfaces, memory size, availability). Each kind and series (identified through the applicable part number) of the used components provides its own Usage Domain Definitions.
Figure 44 – Usage Domain Definitions

Figure 45 – Usage Domain Definitions - Cabinets

A.3  IMACT WORKFLOW
The following figure shows the typical workflow with IMACT (note that iterating several sub-workflows may be necessary):
1. **Project Preparation**
   This step includes the setup of the project and the global properties (e.g. enumerations, roles, users, etc.). This is done manually. The project global properties are the outputs of this step.

2. **Capture and import Hosted Function Requirements**
   This step collects the resource requirements from all hosted functions, imports them into IMACT and makes adjustments if necessary. IMACT can import HFRQs as XML files to the IMACT database. The manual adjustments can be made in the graphical editor within IMACT.
3. Capture and import Usage Domains
   This step collects the usage domain definitions from all IMA modules, imports them into
   IMACT and makes adjustments if necessary. IMACT can import UDs as XML files to the
   IMACT database. The manual adjustments can be made in the graphical editor within
   IMACT.

4. IMA Sizing
   This step determines the amount and type of all IMA modules (i.e. cabinets, IPMs, RDCs
   and AFDX switches) necessary to satisfy all HFRQs including margins and makes
   adjustments if necessary. IMACT performs these calculations automatically. The manual
   adjustments can be made in the graphical editor within IMACT. The IMA module (RDC,
   IPM and AFDX switch) elements in the IMACT database are the outputs of this step.

5. Capture and import System Function Requirements
   This step collects the SFRQs from all IMA platform system functions, imports them into
   IMACT and makes adjustments if necessary. IMACT can import SFRQs as XML files to
   the IMACT database. The manual adjustments can be made in the graphical editor within
   IMACT.

6. Topology
   This step determines the physical topology of all IMA modules, connects all equipment to
   power supplies and AFDX switches, makes adjustments if necessary and analyzes the
   topology. The topology generation is performed automatically by IMACT outputting the
   physical location of the IMA modules and their connection to power buses and AFDX
   switches.
   Then, manual adjustments can be made in the graphical editor within IMACT.
   Afterwards, the physical topology can be automatically analyzed by IMACT, considering
   location, AFDX connection without conflicts and requirement-matching power supply
   connections.

7. RDC and Switch Resources Allocation
   This step allocates the RDC resources (including IO and RDC functions) to hosted functions
   and connects AFDX switch ports to AFDX end systems and other AFDX switches. This
   allocation and connection is performed automatically by IMACT, with regard to IO resource
   requirements, RDC function requirements, the physical location of controllers connected to
   the IO requirements, availability/integrity/separation requirements and spare margins.
   Then, manual adjustments can be made in the graphical editor within IMACT.
   Afterwards, the RDC and switch allocations can be automatically analyzed by IMACT,
   considering satisfaction of all RDC/switch requirements, traceability of all RDC elements to
   requirements, RDC/switch resource spare margins, resource allocation within limits of
   RDC/switch usage domain, instantiation of defined controllers and traceability to
   requirements, RDC resource separation/aggregation/availability/integrity requirements, and
   zone equality of connected RDCs and controllers.

8. IPM Resources Allocation
   This step allocates the IPM resources, including CPU time and memory. This allocation is
   performed automatically by IMACT, with regard to application memory and CPU time
   resource requirements, partition health monitoring requirements,
availability/integrity/separation requirements, and globally defined spare margins. Then, manual adjustments can be made in the graphical editor within IMACT. Afterwards, the IPM resource allocation can be automatically analyzed by IMACT, considering satisfaction of IPM resource requirements by IPM elements, traceability of all instantiated IPM elements to requirements, IPM resource spare margins, and resource allocation within limits of IPM usage domain.

9. Dataflow generation
This step defines the mapping of parameters, generation of AFDX communication elements (messages, ports and virtual links) and generation of IPM APEX communication ports. First, the mapping between system parameters and global parameters is done manually. This is to determine the dataflow expected by the hosted functions. Then, the communication elements (AFDX/APEX) are generated automatically by IMACT, with regard to parameter routing (through controllers, RDCs and applications), grouping of parameters into functional data sets and messages, grouping of messages into virtual links, and routing of virtual links through the AFDX network. After this, manual adjustments can be made in the graphical editor within IMACT. Finally, the dataflow can be automatically analyzed by IMACT, considering consistency of all communication elements (signals, functional data sets, messages, ports, virtual links), conformance of AFDX timing with ARINC standard and usage domain, satisfaction of defined exchanges in the parameter mapping, of HFRQ latency requirements, and of integrity/availability/separation requirements.

10. Export
This step exports the configuration of all IMA modules and AFDX end systems and exports the ICDs of IMA modules. IMACT can export this data as XML files based on the content from the IMACT database.

A.4 IMACT PARTITIONING

IMACT incorporates the principle of partitioning, particularly “time partitioning” and “space partitioning”.

The Integrated Processing Modules (IPMs) are the main components within IMACT to configure partitioning aspects. They manage resources in a manner sufficient to support at least one application. The following is realized through IPMs:

Application partitions, to run multiple applications, by space partitioning (also see section A.3 step 8). Different memory regions to be used by the partitions, hardware resources, communication ports and access details are defined.

Scheduling of a partition/application, i.e. CPU time with duration and period parameters, by time partitioning (also see section A.3 step 8).

Related to the partitioning topic, IMACT includes generators and analyzers:

The Application Partition Allocation Generator distributes the applications on the respective IPM. Conditions to be met are e.g. separation conditions, aggregation conditions, maximum number of applications per IPM, memory and scheduling compatibility, availability and integrity requirements.
The Application Partition Allocation Analyzer verifies the correct distribution of the applications on the IPMs. This is performed on different levels of the IMA system: IMA system level, cabinet level and IPM level. The latter considers all the criteria also respected during automatic generation (see paragraph above).

The analyzer at IMA system level checks that all applications are allocated on the IPMs, and respect the side/zone aggregation and separation conditions with other applications and with controllers and RDC interfaces.

The analyzer at cabinet level checks the cabinet separation and aggregation conditions of the applications on the respective cabinet.

More about IMA related partitioning, as well as safety and security aspects are detailed in [75] and [76].

### A.5 IMACT Outputs

Based on the provided input IMACT is able to generate configuration files for the hardware components utilized by the IMA system. For each used configurable hardware component a configuration file based on its service to be provided is created. One configuration file may be applicable for multiple hardware components of similar types, the actual type is specified via pin programming.

IMACT provides configuration files for AFDX switches, IPM partitions, controllers and RDC end systems.

**Figure 47 – IMACT Configuration Output**

IMACT is also able to generate the network layout description and the interface description. The network layout description specifies the internal wiring of the IMA system. The interface description specifies the used interface types, settings of the interfaces as well as details of the communication on these interfaces, i.e. message structures and the contained signals sent through the interfaces.
IMACT provides the Interface Control Document (ICD) and the network topology as system outputs.

A.6 CONCLUSION

The Integrated Modular Avionics Configuration Tool described in the chapters above enables the user to model IMA systems, to generate configurations for the components of the modelled system, and to perform several analyses on the modelled system. The tool addresses safety and security topics, e.g. DO178B, by applying the technique of partitioning (according to ARINC653). For more information refer to [75] and [76].