Collaborative Large-scale Integrating Project

Open Platform for EvolutioNary Certification Of Safety-critical Systems

Baseline for the Compositional Certification Approach

D5.1

<table>
<thead>
<tr>
<th>Work Package:</th>
<th>WPS: Compositional Certification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dissemination level:</td>
<td>PU</td>
</tr>
<tr>
<td>Status:</td>
<td>Final</td>
</tr>
<tr>
<td>Date:</td>
<td>July 5th, 2012</td>
</tr>
<tr>
<td>Responsible partner:</td>
<td>Bart Jacobs (DNV - inspearit)</td>
</tr>
<tr>
<td>Contact information:</td>
<td><a href="mailto:Bart.jacobs@inspearit.com">Bart.jacobs@inspearit.com</a></td>
</tr>
</tbody>
</table>

PROPRIETARY RIGHTS STATEMENT
This document contains information, which is proprietary to the OPENCOSS Consortium. Neither this document nor the information contained herein shall be used, duplicated or communicated by any means to any third party, in whole or in parts, except with prior written consent of the OPENCOSS consortium.
Contributors

<table>
<thead>
<tr>
<th>Names</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Georgio Taglifelli, Vincenzo Manni, Andrea Felicetti</td>
<td>RINA Services SpA</td>
</tr>
<tr>
<td>Joost Gabriels</td>
<td>Eindhoven University of Technology</td>
</tr>
<tr>
<td>Alessandra Martelli</td>
<td>Intecs</td>
</tr>
<tr>
<td>Jose Luis de Vara, Sunil Nair, Mehrdad Sabetzadeh</td>
<td>Simula Research Laboratory</td>
</tr>
<tr>
<td>Daniela Cancila</td>
<td>ATEGO France</td>
</tr>
<tr>
<td>Bart Jacobs</td>
<td>Inspeart (before DNV ITGS)</td>
</tr>
<tr>
<td>Bernhard Sputh</td>
<td>ALTREONIC</td>
</tr>
<tr>
<td>Philippa Conmy, Oleg Lisagor</td>
<td>University of York</td>
</tr>
<tr>
<td>Fulvio Tagliabò, Alberto Melzi</td>
<td>Centro Ricerche Fiat S.C.p.A.</td>
</tr>
<tr>
<td>Cedric Chevreles</td>
<td>THALES Avionics</td>
</tr>
<tr>
<td>Laurent de la Beaujardiere, Fabien Belmonte</td>
<td>Alstom Transport</td>
</tr>
</tbody>
</table>

Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>V0.1</td>
<td>2012-02-15</td>
<td>T.o.C.</td>
</tr>
<tr>
<td>V0.2</td>
<td>2012-02-24</td>
<td>First draft version</td>
</tr>
<tr>
<td>V0.3</td>
<td>2012-03-14</td>
<td>Second draft version</td>
</tr>
<tr>
<td>V0.4</td>
<td>2012-03-28</td>
<td>Third draft version</td>
</tr>
<tr>
<td>V0.5</td>
<td>2012-04-04</td>
<td>Forth draft version, new structure</td>
</tr>
<tr>
<td>V0.6</td>
<td>2012-04-13</td>
<td>Several sections added</td>
</tr>
<tr>
<td>V0.7</td>
<td>2012-04-20</td>
<td>Suggestions after intermediate internal review</td>
</tr>
<tr>
<td>V0.8</td>
<td>2012-04-24</td>
<td>Revised structure, comments solved, new contributions</td>
</tr>
<tr>
<td>V0.9</td>
<td>2012-05-02</td>
<td>New contributions, figures and tables added</td>
</tr>
<tr>
<td>V0.10</td>
<td>2012-05-09</td>
<td>References renewed, sections added</td>
</tr>
<tr>
<td>V0.11</td>
<td>2012-05-15</td>
<td>All contributions in draft</td>
</tr>
<tr>
<td>V0.12</td>
<td>2012-06-08</td>
<td>Consistency and alignment</td>
</tr>
<tr>
<td>V0.13</td>
<td>2012-06-18</td>
<td>Ready for peer review</td>
</tr>
<tr>
<td>V0.14</td>
<td>2012-07-02</td>
<td>Ready for PB review</td>
</tr>
<tr>
<td>V1.0</td>
<td>2012-07-05</td>
<td>Approved by PB</td>
</tr>
</tbody>
</table>


## TABLE OF CONTENTS

1 **Introduction** ........................................................................................................................................ 10  
1.1 Compositional certification specific definitions ......................................................................................... 10  
1.2 Document structure ........................................................................................................................................ 12  
1.3 Technical context and objectives for a contract-based approach for compositional certification ........................................................................................................................................ 12  
1.4 Relation to other work packages and deliverables .......................................................................................... 15  
1.4.1 Relation to High level requirements (WP2) .......................................................................................... 15  
1.4.2 Relation to Platform Integration and Validation (WP3) ........................................................................ 15  
1.4.3 Relation to Common Certification Language (WP4) .............................................................................. 16  
1.4.4 Relation to Chain of evidence (WP6) .................................................................................................. 18  
1.4.5 Relation to Transparent Certification and Compliance-aware Process (WP7) ........................................... 19  

2 **Results of the baseline survey on compositional certification** .............................................................................. 20  
2.1 Motivations for reusing components .......................................................................................................... 20  
2.2 Challenges of reusing components ............................................................................................................ 20  
2.3 Experience with reuse from OPENCOSS partners .......................................................................................... 21  
2.4 Experience with compositional or modular certification ................................................................................. 21  

3 **Safety certification** ................................................................................................................................. 22  
3.1 Safety Certification approaches ................................................................................................................. 22  
3.2 Modularity and compositionality in system design ....................................................................................... 24  

4 **Purpose of safety standards from reuse perspective** ....................................................................................... 26  
4.1 Automotive .................................................................................................................................................. 27  
4.2 Avionics/aviation ........................................................................................................................................ 30  
4.2.1 Civil Aviation ARP4754 & ARP4761 .................................................................................................. 32  
4.3 Railway .................................................................................................................................................... 34  
4.3.1 Safety Approval Process .................................................................................................................... 36  
4.3.2 Dependency between Safety Approvals .............................................................................................. 38  
4.4 Other domains: Defence ............................................................................................................................. 38  

5 **Overview of the state of industrial practice** .................................................................................................. 40  
5.1 Automotive ................................................................................................................................................. 40  
5.2 Avionics/aviation ........................................................................................................................................ 41  
5.2.1 Civil aviation ........................................................................................................................................ 41  
5.3 Rail ........................................................................................................................................................... 44  
5.4 Other domains: Defence ............................................................................................................................. 47  

6 **State-of-the-art** ......................................................................................................................................... 49  
6.1 Design modularity: Standardised architectures, interfaces and partitioning ......................................................... 49  
6.1.1 Integrated Modular Avionics ................................................................................................................. 49  
6.1.2 AUTOSAR ........................................................................................................................................... 51  
6.2 Design modularity: Design-by-contract ...................................................................................................... 51  
6.3 Compositional safety cases .......................................................................................................................... 54  
6.3.1 Key principles of modular safety cases ............................................................................................... 55  
6.3.2 From modular safety cases to compositional certification ........................................................................ 61  
6.4 Other research work relevant for WP5 ........................................................................................................ 62  
6.5 Compositional Certification tooling ........................................................................................................... 63  

7 **Overview of projects and initiatives related to compositional modular certification** .................................. 65  

8 **Conclusions** ............................................................................................................................................... 67
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1</td>
<td>Summary of findings</td>
<td>67</td>
</tr>
<tr>
<td>8.2</td>
<td>Areas requiring further R&amp;D work</td>
<td>69</td>
</tr>
<tr>
<td>8.2.1</td>
<td>Matching explicit context and assumptions</td>
<td>69</td>
</tr>
<tr>
<td>8.2.2</td>
<td>Implicit context and assumptions in safety case evidence</td>
<td>69</td>
</tr>
<tr>
<td>8.2.3</td>
<td>Reflecting modular structure in confidence and compliance arguments</td>
<td>70</td>
</tr>
<tr>
<td>8.2.4</td>
<td>Trade-offs between compositional certification and monolithic safety case development</td>
<td>70</td>
</tr>
<tr>
<td>8.2.5</td>
<td>Industrial evaluation</td>
<td>71</td>
</tr>
<tr>
<td>8.3</td>
<td>Next Steps</td>
<td>71</td>
</tr>
<tr>
<td>9</td>
<td>Appendix A</td>
<td>75</td>
</tr>
<tr>
<td>10</td>
<td>Appendix B</td>
<td>76</td>
</tr>
<tr>
<td>10.1</td>
<td>FP Research Projects</td>
<td>76</td>
</tr>
<tr>
<td>10.2</td>
<td>ARTEMIS Joint Undertaking projects</td>
<td>80</td>
</tr>
<tr>
<td>10.3</td>
<td>ITEA R&amp;D Projects</td>
<td>83</td>
</tr>
<tr>
<td>10.4</td>
<td>Other initiatives</td>
<td>84</td>
</tr>
</tbody>
</table>
# Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARP</td>
<td>Aerospace Recommended Practice</td>
</tr>
<tr>
<td>ARTEMIS</td>
<td>Advanced Research &amp; Technology for EMbedded Intelligence and Systems</td>
</tr>
<tr>
<td>ASIL</td>
<td>Automotive Safety Integrity Level</td>
</tr>
<tr>
<td>ASSERT</td>
<td>Automated proof-based System and Software Engineering for Real-Time systems</td>
</tr>
<tr>
<td>AUTOSAR</td>
<td>AUTomotive Open System ARchitecture</td>
</tr>
<tr>
<td>CAE</td>
<td>Claims, Arguments and Evidence</td>
</tr>
<tr>
<td>CCA</td>
<td>Common Cause Analysis</td>
</tr>
<tr>
<td>CCL</td>
<td>Common Certification Language</td>
</tr>
<tr>
<td>CENELEC</td>
<td>Comité Européen de Normalisation Electrotechnique (European Committee for Electrotechnical Standardization)</td>
</tr>
<tr>
<td>CESAR</td>
<td>Cost-efficient methods and processes for safety relevant embedded systems</td>
</tr>
<tr>
<td>CHESS</td>
<td>Composition with Guarantees for High-integrity Embedded Software Assembly</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial Of-The-Shelve</td>
</tr>
<tr>
<td>DECOs</td>
<td>Dependable Embedded Components and Systems</td>
</tr>
<tr>
<td>DSR</td>
<td>Derived Safety Requirement</td>
</tr>
<tr>
<td>EVOLVE</td>
<td>Evolutionary Validation, Verification and Certification</td>
</tr>
<tr>
<td>FHA</td>
<td>Functional Hazard Assessment</td>
</tr>
<tr>
<td>FP</td>
<td>Framework Programme</td>
</tr>
<tr>
<td>GA</td>
<td>Generic Application</td>
</tr>
<tr>
<td>GENESYS</td>
<td>Generic Embedded System Platform</td>
</tr>
<tr>
<td>GP</td>
<td>Generic Product</td>
</tr>
<tr>
<td>GSN</td>
<td>Goal Structuring Notation</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>IAWG</td>
<td>Industrial Avionics Working Group</td>
</tr>
<tr>
<td>IMA</td>
<td>Integrated Modular Avionics</td>
</tr>
<tr>
<td>ITEA</td>
<td>Information Technology for European Advancement</td>
</tr>
<tr>
<td>MAENAD</td>
<td>Model-based Analysis &amp; Engineering of Novel Architectures for Dependable Electric Vehicles</td>
</tr>
<tr>
<td>MBA</td>
<td>Model-Based Engineering</td>
</tr>
<tr>
<td>MBAT</td>
<td>Combined Model-based Analysis and Testing of Embedded Systems</td>
</tr>
<tr>
<td>MISSA</td>
<td>More Integrated Systems Safety Assessment</td>
</tr>
<tr>
<td>MODSafe</td>
<td>Modular Urban Transport Safety and Security Analysis</td>
</tr>
<tr>
<td>OPENCOSS</td>
<td>Open Platform for EvolutioNairy Certification Of Safety Systems</td>
</tr>
<tr>
<td>PSSA</td>
<td>Preliminary System Safety Assessment</td>
</tr>
<tr>
<td>RAMS</td>
<td>Reliability, Availability, Maintainability and Safety</td>
</tr>
<tr>
<td>RECOMP</td>
<td>Reduced Certification Costs Using Trusted Multi-core Platforms</td>
</tr>
<tr>
<td>SafeCer</td>
<td>Safety Certification of Software-Intensive Systems with Reusable Components</td>
</tr>
<tr>
<td>SEooC</td>
<td>Safety Element out of Context</td>
</tr>
<tr>
<td>SPEEDS</td>
<td>SPEculative and Exploratory Design in Systems engineering</td>
</tr>
<tr>
<td>SRAC</td>
<td>Safety Related Application Conditions</td>
</tr>
<tr>
<td>SSA</td>
<td>System Safety Analysis</td>
</tr>
<tr>
<td>SysML</td>
<td>System Modeling Language</td>
</tr>
<tr>
<td>V&amp;V</td>
<td>Verification and Validation</td>
</tr>
<tr>
<td>WP</td>
<td>Work Package</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Modularity in the same domain</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>Modularity with cross domain re-use of certification data (OC stands for OPENCOSS)</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>CCL translation paradigm</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>Relationship between assumptions and SEooC development, from ISO 26262 - part 10</td>
<td>29</td>
</tr>
<tr>
<td>5</td>
<td>Certification processes (Eveleens, 2007)</td>
<td>31</td>
</tr>
<tr>
<td>6</td>
<td>IMA Development Guidance and Certification Considerations (Eveleens, 2006)</td>
<td>32</td>
</tr>
<tr>
<td>7</td>
<td>Simplified ARP4754a safety lifecycle, and relationship with the scope of DO178B</td>
<td>33</td>
</tr>
<tr>
<td>8</td>
<td>Safety Acceptance and approval process taken from ENS0129</td>
<td>37</td>
</tr>
<tr>
<td>9</td>
<td>Examples of dependencies between Safety Cases / Safety Approval</td>
<td>38</td>
</tr>
<tr>
<td>10</td>
<td>Simplified view of requirements flow-down in ARP4754/ARP4761 process</td>
<td>43</td>
</tr>
<tr>
<td>11</td>
<td>Example of dependencies between five aircraft systems</td>
<td>44</td>
</tr>
<tr>
<td>12</td>
<td>Illustration of Cross Acceptance</td>
<td>46</td>
</tr>
<tr>
<td>13</td>
<td>IMA three-layer stack</td>
<td>50</td>
</tr>
<tr>
<td>14</td>
<td>Components and contracts</td>
<td>52</td>
</tr>
<tr>
<td>16</td>
<td>The contact-based approach developed during the ASSERT project</td>
<td>53</td>
</tr>
<tr>
<td>17</td>
<td>Match between contracts</td>
<td>53</td>
</tr>
<tr>
<td>18</td>
<td>Safety Case Module interface</td>
<td>56</td>
</tr>
<tr>
<td>19</td>
<td>Simple example of a GSN module with external references</td>
<td>57</td>
</tr>
<tr>
<td>20</td>
<td>Simple Safety Case Architecture Example</td>
<td>58</td>
</tr>
<tr>
<td>21</td>
<td>Sun’s Reference Safety Case Architecture for ARP4754a</td>
<td>59</td>
</tr>
<tr>
<td>22</td>
<td>Generic Contract for Safety Case Contract (Fenn &amp; al, 2007)</td>
<td>61</td>
</tr>
</tbody>
</table>
List of Tables

Table 1: Key Safety Standards in Automotive, Civil Aviation and Railway domains ........................................ 23
Table 2: Suggestions for tooling including description for compositional certification .................................. 64
Table 3: Initiatives including possible useful input for compositional modular certification ......................... 65
Table 4: Questions related to WP5 in the survey that has been conducted with OPENCOSS partners ........ 75
Executive Summary

This document (D5.1) is the first deliverable of OPENCOSS WP5. This WP aims to define a compositional/modular, contract-based certification approach compatible with the Common Certification Language (CCL) to be specified in WP4. Therefore, D5.1 establishes the foundations for the main part of the future work in WP5 and serves as scoping of the WP.

D5.1 will mainly serve as input for D5.2, in which detailed requirements for the compositional certification approach infrastructure will be specified. A part of these requirements can be discovered from D5.1 and from the insights it gives into what work is currently lacking in this area. In addition, the deliverable provides valuable input for other OPENCOSS WPs and deliverables by means of the detailed review of the state of the art, the outline of relationships between WP5 and other WPs, and the overview of related projects and initiatives.

To this end, D5.1 presents: (1) the relationship of WP5 with other OPENCOSS WPs; (2) information from OPENCOSS partners regarding compositional certification; (3) an overview of safety certification, describing the purpose of safety standards from a reuse perspective; (4) the state of the practice by which safety-critical industries conform to safety standards from a compositional perspective; (5) existing work on compositional certification; and (6) related projects and initiatives for compositional certification.

A key point is to understand how to capture the properties of each system component (and evidence of those properties) as a contract, and how to combine and match the contracts as part of a compositional certification approach. A further challenge is the identification of the necessary contractual information and how to capture it. Finally, emergent properties or unexpected interactions which may arise during integration can be assessed using the contracts.

As a result, D5.1 provides insights into existing work, approaches and practices. It summarizes the objectives for a compositional certification approach, including limitations and challenges. From there, it will be clear what is missing and what the OPENCOSS project can improve.

The main findings and results provided by D5.1 are:

- There is a strong correlation between WP5 (compositional certification), WP4 (CCL) and WP6 (chain of evidence);
- Successful compositional certification can be supported by standardised, well-defined software architectures;
- The focus of WP5 is on development of defensible, consistent and coherent safety cases rather than merely demonstration of compliance with low-level requirements of prescriptive safety standards;
- The focus of WP5 is on modular safety cases instead of monolithic artefacts (safety case modules, safety case architectures, safety case contracts);
- Information hiding for components, hierarchy of components and interfaces between components are key in compositional certification;
- Safety case architecture forms a basis for compositional certification (high cohesion, low coupling, module interfaces, modular dependencies);
• Safety case contracts record interdependencies between assurance arguments, while software contracts record dependencies between components;
• Safety case modules can only be properly composed if assumptions (and other contextual information) associated with evidence cited by different modules are compatible;
• There is only a small set of tools available for compositional certification according to the consortium, more common tools must be analysed (see Section 6.5);
• Several projects and initiatives – such as SafeCer and IAWG – relevant to WP5 must be investigated further (see Section 7).

It is important to note that even within the tightly regulated industrial domains and within the jurisdiction of prescriptive safety standards, a safety engineers’ judgement is indispensable for compositional certification. Whilst the OPENCOSS project and WP5 aim to provide a framework, guidance and tools to facilitate and support compositional certification of safety-critical systems, the project will not attempt to fully automate safety assurance, certification, and safety case composition/development or even review tasks.
1 Introduction

Safety assurance and certification are amongst the most expensive and time-consuming tasks in the development of safety-critical embedded systems. European innovation and productivity in this market is curtailed by the lack of affordable (re)certification approaches. Major problems arise when evolutions to a system entail reconstruction of the entire body of certification arguments and evidence. Further, market trends strongly suggest that many future embedded systems will be comprised of heterogeneous, dynamic coalitions of components. As such, they will have to be built and assessed according to numerous standards and regulations. Current certification practices (e.g., traditional, monolithic and standards-based approaches) will be prohibitively costly to apply to this kind of embedded systems (Rushby, 2007).

The OPENCOSS project aims to devise a common certification framework that spans different vertical markets for railway, avionics/aviation and automotive industries, and to establish an open-source safety certification infrastructure. The ultimate goal of the project is to bring about substantial reductions in recurring safety certification costs, and at the same time increase product safety through the introduction of more systematic certification practices. Both will boost innovation and system upgrades considerably.

This project-wide goal will be supported by WP5 through provision of a compositional certification framework. The envisaged framework will seek to enable specification and exchange of the certification assets related to components of safety-critical systems and effective utilisation of these assets in certification of safety critical systems. Furthermore the framework will facilitate reuse of the certification assets when components themselves are reused (whether within or across vertical markets) while limiting the amount of re-certification work that is required following a change to the system design.

This first deliverable of WP5 serves mainly as input for D5.2 (low-level requirements for compositional certification). In order to discover these requirements, this document will describe the state-of-the-art for compositional certification, from an industrial as well as from a scientific perspective. This document will provide input for a gap analysis, to understand what is lacking in the area of compositional certification.

1.1 Compositional certification specific definitions

The terms component or module are variously used in different sub-fields of Systems and Software Engineering with broadly similar meaning of "a part of the whole" (OED). To maintain clarity within this document as far as possible we use the term "component" to refer to part of the system or its various models. The definition of this term is intentionally broad and covers parts of the system that range from major sub-systems to the smallest engineered items. Unless explicitly stated otherwise, the term "module" is typically used in the document as a short-hand for "safety case module" - a well-defined part of the safety case that may or may not be related to a system component.

WP5 is mainly about compositional certification. The following topics play an important role within compositional certification: certification, component(s), evidence and claim(s). Below, we provide basic working definitions for those topics that are used throughout this document.
Certification: Legal recognition by a certification authority that a product, service, organization or person complies with some predetermined requirements. Such certification comprises the activity of technically checking the product, service, organization or person and the formal recognition of compliance with the applicable requirements by issue of a certificate, license, approval or other documents as required by national laws and procedures. In particular, certification of a product involves: (a) the process of assessing the design of a product to ensure that it complies with a set of standards applicable to that type of product so as to demonstrate an acceptable level of safety; (b) the process of assessing an individual product to ensure that it conforms with the certified type design; (c) the issuing of a certificate required by national laws to declare that the product complies with relevant standards in accordance with items (a) or (b) above. The definition is from (DO-178B - Software Considerations in Airborne Systems and Equipment Certification, 1992) and (DO-297: Integrated Modular Avionics (IMA), 2007). OPENCOSS is primarily concerned with the certification of safety critical systems.

Certification data or certification assets: includes both claims and evidence used as part of the certification process, which are produced within the development and assessment of a safety-critical product. Within this document the two terms are used interchangeably. They are also closely related to the notion of “safety case module” (see below).

Claim: An assertion made about the system in question, typically with reference to a safety and/or functional property or behaviour. The full set of claims should form a coherent argument about whether the risk associated with the system has been sufficiently reduced/mitigated or removed. Claims are linked to, and supported by, evidence. Claims identify the adverse consequences to be considered and the degree of risk considered tolerable; evidence comprises the results of analyses, reviews, and tests; and the argument makes the case, based on the evidence, that the claims are satisfied. (Rushby, 2007).

Component: A self-contained part, combination of parts, subassemblies or units, which performs a distinct function of a system. A self-contained hardware or software part, database, or combination thereof that may be configuration controlled. (DO-178B - Software Considerations in Airborne Systems and Equipment Certification, 1992) and (DO-297: Integrated Modular Avionics (IMA), 2007). Note that within this document the term “component” is defined intentionally broadly and may cover parts of the system that range from basic items to major sub-systems. To avoid confusion the term “component” refers to the parts of the system or its design whereas the term “safety case module” (see below) refers to parts of the system safety case.

Evidence: Results of observing, analysing, testing, simulating and estimating the properties of a system that provides the fundamental information from which safety can be inferred.

Safety Case Module: a coherent part of a safety case with a well-defined interface. Defining a safety case module involves defining the objectives, evidence, argument and context associated with one aspect of the safety case.

It should be noted that, whilst some of the research on use of Commercial Off the Shelf (COTS) components in safety-critical application may prove relevant to the research conducted by WP5, the focus of the work
package is on reuse of components that are associated with comprehensive certification data and on composition of such component-specific data into an overall system safety case. Such components are sometimes referred to as “pre-qualified”. However, since this term is used inconsistently by different authors, we avoid it in the remainder of the present report.

1.2 Document structure

The remainder of this chapter describes the context and objectives for a contract-based approach to compositional certification. It clarifies the relationship between such a contract-based approach of compositional certification studied in WP5 and other technical work packages (WP2, 3, 4, 6 & 7) of the project.

Chapter 2 provides a summary of the results from a baseline survey, containing input from most OPENCOSS partners on motivation, challenges, pros and cons and experiences for compositional (modular) certification.

Chapter 3 contains an overview of different approaches for certification between different domains. Furthermore, it contains an overview of the notions of modularity and composability in safety assessment and design.

Chapter 4 describes the purpose of safety standards, and provides more detail on objectives of currently used safety standards within the different target domains of OPENCOSS (automotive, avionics and railway), including objectives of safety standards outside of these three target domains.

Chapter 5 focusses on the different industries within the target domains, in order to understand how industries conform to the safety standards.

Chapter 6 describes the state-of-the-art for compositional certification by introducing a number of “streams” of approaches for design modularity and compositional certification.

Chapter 7 summarizes research projects and initiatives related to compositional (modular) certification.

Chapter 8 provides the conclusions and pulls together the objectives and requirements, existing work and challenges where new research efforts for compositional certification are necessary. It will also scope the further WP5 work.

1.3 Technical context and objectives for a contract-based approach for compositional certification

The objective of WP5 is to define a compositional, contract-based certification approach that is compatible with the CCL to be developed within the OPENCOSS project (WP4). A key point is to understand how to capture each system component’s properties (and evidence of those properties) as a contract, and how to combine and match the contracts as part of a compositional certification approach. A further challenge is
the identification of the necessary contractual information and how to capture it. Finally, contracts should allow assessment of emergent properties or unexpected interactions that may arise during integration.

The term compositional certification refers to the ability to achieve system certification for a safety-critical system that has been built from multiple components, some or all of which have a pre-existing set of certification data (partial or complete), provided by the component supplier. The aim is to reuse this data (within a single domain or even cross-domain), and thus to reduce the time to build, assess and certify the systems. This also implies a reduction in the costs associated with re-verification. Reuse may be within a single domain (Figure 1), or cross-domain (Figure 2) for suitable components.

![Diagram showing modularity in the same domain](image)

**Figure 1: Modularity in the same domain**

Cross-domain reuse is particularly challenging, due to different requirements, hazards and standards across those domains. Therefore, component certification data generated for a particular regulatory context may require a degree of translation and adaptation before it can be used within another domain (as shown for Component A in Figure 2). Hence the need for a CCL that captures these subtle differences and helps us to better understand how and if certification data can be reused. Ultimately, we need to demonstrate that a system is acceptably safe, so great care is needed not to reuse data without due consideration.
Certification data refers here to safety arguments, safety evidence, and any other analysis data related to the integrity of a component. There are three main aspects to compositional certification:

- **Compositional evidence** - evidence (particularly safety evidence) is produced by applying different analysis and testing techniques to a component, to back up what has been described in the contract (e.g. to prove that the execution time is that specified). See also the evidence taxonomy in D6.1. Some types of evidence are easier to compose than others - for example some kinds of testing can be performed per software module, and fault trees are often composable. However, these may be linked to specific system requirements so care is needed to ensure that the properties being demonstrated are still valid in a new context. Other types of evidence may superficially seem easy to compose, but be harder in practice. For example, Worst Case Execution Time (WCET) values can be added together very easily, but values may have been derived using very different techniques (static analysis or probabilistic testing). Other types of evidence are unlikely to be composable (e.g. top down functional failure analysis).

- **Compositional safety arguments** - The safety argument about a single component will capture information about its assumed requirements, and the context within which evidence was derived. It will contain arguments about the evidence, such as the example in the section above for WCET, and why it is believed to adequately show a particular requirement is met. A system safety argument must bring together safety argument fragments from component’s certification data, to make a compelling case for the whole system.
- **Design for composability** - It is important to realise that some system- and software architectures also support compositional certification. For example, Integrated Modular Avionics (IMA) has partitioning mechanisms, to prevent resource contention, thus reducing application interference and making it easier to argue about, and analyse, some components separately.

The challenge of safety-critical systems is to assess not only the certifiability of each component or module, but also its certifiability once it is in an ‘integrated’ state. For example, if the safety argument relies, in part, on reasoning from the properties of components, then the system build process should leave evidence that the system has been built out of the specific versions of each component for which there is evidence that the component has the said properties. Each step in developing the software needs to preserve the chain of evidence on which the argument that the resulting system is dependable will be based. This is the basic principle of a compositional certification approach.

### 1.4 Relation to other work packages and deliverables

OPENCOSS technical work has been divided into several WPs; each associated with specific objectives and planned results. Nonetheless, the WPs are not completely independent. Some WPs must fulfil the needs of others so that the OPENCOSS platform is successfully developed. This section analyses relationships between WP5 and other WPs.

The following subsections discuss the relationships between WP5 and every other relevant WP, and determine the needs that must be fulfilled and aspects that must be considered during WP5 development. The relationships with WP1 (Use Case Specification and Benchmark) are not analysed here, because they provide mainly input for WP2 (we refer interested reader to OPENCOSS D6.1). The relationships with WP8 (Standardization and Community Building), WP9 (Dissemination, Training and Exploitation) and WP10 (Project Management) are not analysed here, because these WPs relate generally to managerial aspects of the project, rather than to the technical work. It must be noted that this is an initial, preliminary analysis made from the perspective of WP5. It will require further study, refinement and harmonization with the rest of the WPs.

#### 1.4.1 Relation to High level requirements (WP2)

OPENCOSS aims to define and implement an intelligent, automated, and highly customizable safety certification management infrastructure, integrated into the development processes and with existing development and safety assurance tools. In order to understand what this means for the design and implementation of the platform, and also for the architecture-based relation with WP5, the problem and solution strategy must be examined. The result of this examination is captured in WP2, which defines high-level requirements which give an overview of the functional and non-functional requirements for the OPENCOSS platform.

One of the main objectives of OPENCOSS, and the focus of WP5, is the provision of an approach to compositional certification. The goal of WP2 is to decompose this and other objectives of the project into a set of high-level requirements as well as, where appropriate, allocate those requirements to the technical...
work packages (WP4, WP6 and WP7). An additional goal of WP2 is to ensure that low-level requirements produced by WP5 are consistent with the requirements elicited by other technical work packages.

1.4.2 Relation to Platform Integration and Validation (WP3)

The aim of OPENCOSS WP3 is the integration and validation of the platform that will be developed in WP4 to WP7. For this purpose WP5 will work with WP3 in order to coordinate the interfaces between WP5 and WP4, WP6, and WP7. Furthermore, the data for the industrial use cases, which are gathered in T3.1 and reported in D3.1, will be integrated into the work done in WP5, where applicable to WP5. This data can be used for requirements discovery in the scope of WP5.

1.4.3 Relation to Common Certification Language (WP4)

Among other objectives, the CCL research aims at defining a means for specifying safety cases in order to meet the goals of OPENCOSS and the realization of an editor for this language. A challenge is to capture both the safety case approach and the standards-based approach. The editor should be part of a larger OPENCOSS platform with which stakeholders can, amongst others, reuse artefacts of existing safety cases for other safety cases (e.g. in other domains or contexts). The relation between the CCL and a compositional certification approach is mainly in the compatibility of certification data.

The goal of WP4 is to define a common conceptual and notational framework for specifying certification assets in OPENCOSS. This framework is known within the project as the CCL. The language will aim to synthesize the core concepts of assurance and compliance management from the different target domains and the relationships between them into a metamodel and natural language definition. This ensures capture of the essential, generic, meaning of ‘what it means for a product to be assured’, independent of any particular domain-specific standard or guidance document. From this “conceptual synthesis”, mappings will be made to instantiations of the core concepts in domain-specific and standard-specific ‘dialects’, also in WP4. Thus, reuse of assets (typically, evidence artefacts, requirements and argument fragments) between the target domains (or between different standards within a single domain) is conceived as an act of translation between models, through the conceptual metamodel provided by the CCL. An example is shown in Figure 3.
Figure 3 shows the process of translation required for the successful reuse of an asset specified according to the norms of a given domain (for example, the automotive standard ISO 26262) in a domain governed by another domain-specific standard (for example, the avionics standard DO-178B). The CCL metamodel (represented in Figure 3 by a cloud) provides generic definitions of relevant common concepts. Two mappings are also provided: one between the current usage in the first domain and the generic interpretation provided by the CCL, and one between the usage in the second domain and the CCL. The original asset is passed to the domain A mapping, which results in a transformation into an interpreted asset in terms of the CCL. This interpretation is then passed to the domain B-CCL mapping, and via that the asset is translated into a form reflecting the norms of domain B.

Within this document, the terms ‘asset’ or ‘certification data’ are used to mean any of the reusable certification artefacts which are produced within the development and assessment of a safety-critical system. In general terms, this implies any of the artefacts which might be used in the demonstration of the safety of the system – either in a safety argument-based approach or a simple compliance checklist -, including requirement statements at different levels of detail, designs and specifications, analysis results, test results, test plans and safety argument fragments (claims, lines of reasoning etc.).

In WP5, the challenge is to provide a means to facilitate the reuse of certification assets within a safety argument framework. The CCL will provide support for this work by providing a clearly-defined, standardised lexicon in which the semantics of individual requirements, argument claims and elements of reasoning can be expressed. This will facilitate the development of well-scoped, generic safety argument modules and interface specifications, and will de-risk the composition of argument and evidence artefacts originally defined with respect to the requirements of diverse domains and/or standards, by making the underlying concepts at work in the assurance and argument processes clear.

It should be noted, however, that evidence and arguments produced for the assurance of a safety-critical system are developed in response to very specific requirements concerning the safety aspects of the
system. These requirements are necessarily derived from hazard analysis carried out on the system, at the level of the system requirements. Safety-related functions are likely to be partitioned across multiple components, and the safe operation of the system as a whole will rely on complex assumptions and guarantees about the behaviour of the aggregated components in a certain context. It will be the task of WP5 to clarify these assumptions and component interdependencies in such a way as to make explicit to what extent component reuse is possible in isolation from the original system context. The CCL to be developed in WP4 can provide some support for this, by providing a means by which the system-level requirements and assumptions can be expressed as generic concepts. The satisfaction of dependencies on the integration of the component can be queried and assured in the reuse context, in terms of the degree of ‘conceptual mapping’ between the translated assets.

Finally, it should be noted that the CCL will include definitions for generic concepts relating to the characteristics of evidence. It is likely that the argument approach adopted in WP5 will need to make explicit the required characteristics of evidence assets, in terms of the role they play in supporting argument claims. By expressing these characteristics in general terms – using the definitions of concepts provided by the CCL - the process of assuring safe reuse of the evidence (in terms of the assurance of specific required features covered by claims in an argument module) can be made considerably easier.

1.4.4 Relation to Chain of evidence (WP6)

OPENCOSS WP6 is concerned with defining the part of the OPENCOSS platform that will support an evolutionary chain of certification evidence. A chain of certification evidence is a series of pieces of evidence that are related (e.g., the agent that has created a requirements specification, the test derived from the requirements, the agent that executed the tests, the report where the test results are documented, etc.). By evolutionary, we mean that a chain of evidence can suffer changes (e.g., a requirement is changed), and thus evolve. As a result of the change, the chain of evidence might not any longer be adequate for safety certification (e.g., the related test cases might have to be updated). Therefore, WP6 needs to provide the necessary methods and supporting tools for the management of evidence used in the safety certification of critical systems, and also needs to pay particular attention to situations in which the evidence changes or evolves. When evidence changes, it must be possible to determine whether the set of safety certification evidence for a system is still adequate or if new evidence and, therefore, re-execution of certification-related activities are necessary. In addition to dealing with evolutionary evidence issues, the WP6 infrastructure will be responsible for evidence storage and evidence lifecycle management.

The WP6 infrastructure will provide the WP5 infrastructure with the necessary information to determine if evidence (or argument) composition is possible. Therefore, it must be determined what kind and quantity of information the WP5 infrastructure will require to determine if composition is adequate. It must be determined what information characterizes a component and a contract, the basis for evidence composition.

The WP5 infrastructure will have to inform the WP6 infrastructure about the positive or negative results with regard to the possibility of composition. If the results are negative, then the WP5 infrastructure will also have to indicate the reason(s), whether the results could probably become positive, and how. Both WP
infrastructures will also need a common evidence schema to exchange evidence, which will have to match the CCL.

1.4.5 Relation to Transparent Certification and Compliance-aware Process (WP7)

WP7 aims at defining a safety certification management infrastructure to support the certification process. This process will be interwoven with the development and safety assurance processes by allowing developers to assess where they are with respect to their duties to conform to safety practices and standards, and still to motivate them to see the effective progress of the work and level of compliance. This WP will define part of the OPENCOSS safety certification management infrastructure concerned with the process view of certification (specification and execution). In particular WP7 will:

- Identify metrics for the certification and safety assurance processes with the pursuit of dependability as a balancing of costs and benefits and a prioritization of risks.
- Design and implement a set of OPENCOSS platform services for certification life-cycle support, standards-compliance awareness, traceability management of certification requirements, and event triggering infrastructure for certification compliance.

WP7 and WP5 have strong relations since the compositional/modular, contract-based certification approach developed in WP5 must be compatible with the certification process supported by the safety certification management infrastructure of WP7 and vice versa.
2 Results of the baseline survey on compositional certification

For all technical WPs, a survey has been conducted with OPENCOSS partners in order to gather input about projects, guidelines, tools and techniques related to the different technical work packages. The survey questions can be found in Appendix A, Table 4.

Reuse of components has (expected) advantages, as well as (expected) disadvantages. In this section, we present a summary of motivations for reusing components, followed by expected challenges of reusing components. It is important to stress that results are presented as reported by the project partners.

2.1 Motivations for reusing components

Partners have reported that expected benefits that can be achieved by reusing components, may lead to development efficiency by saving of time and reduction of cost, due to reduction of both the effort of creating the software, testing, debugging and fine-tuning. Such reuse of components can make certification more systematic and manageable, due to better analysis of system integration issues (emergent issues). In some domains, composition of systems from existing certified generic products results in reduction of the re-certification costs and efforts for reused parts.

2.2 Challenges of reusing components

However, in order to achieve such benefits as described within the Section above, a set of challenges should be tackled. Certification and safety assurance could be error prone, while it is difficult to justify the safety of a reused component without providing a full certification kit. There is the risk of different contexts, jeopardising implicit assumptions made about compatibility which compromise system safety. When reused components are not integrated in the overall system development lifecycle, the gap at the boundary between system requirements and generic product features may lead to risks of discovering system integration constraints. Moreover, a component certification data should be complete and consistent, containing consistent evidence characterization, component models and specifications, and complementary data in order to assess the possibilities for reuse in another context. Therefore, component-level arguments should be encapsulated effectively.

The “size” of reusable components should be reduced from generic product to software and hardware modules. Also, potential cost of having to manage a high number of modifications and verifications after reusing should be taken into account, as well as achieved cost advantages that occur over long lifecycle with higher up-front costs. Finally, as different domains use different standards and are subjected to different (national) regulation, cross-domain reuse of components is difficult to achieve in practice.
2.3 Experience with reuse from OPENCOSS partners

It is impossible to accurately quantify the degree of reuse in the three industrial domains directly addressed by the project. Firstly, most partners do not have metrics available, due to confidentiality or due to their role (e.g. researchers, tool providers). Secondly, OPENCOSS consortium does not provide an adequate (statistically significant) sample for drawing general industry-wide conclusions with any degree of confidence. However, a number of partners have suggested that approximately 80% of components of products and applications can be reused in Automotive and Railway domains. Regardless of details of this figure, this provides a strong indication that reuse of components is a significant consideration in development of safety critical systems. Partners have reported that the types of components that can be reused include: hardware units, operating systems, runtime kernels, subsystems, drivers, middleware, processors, software procedures, subsystems, harness, software modules, software libraries, drivers, and interfaces.

2.4 Experience with compositional or modular certification

A component should be accompanied by, on the one hand, evidence concerning the degree of rigour used in its development (e.g. level of testing, reviews, etc.) and, on the other hand, specification of the basic assumptions for component’s use. The challenge is to deal with mismatches in terminology and approach, when comparing cross-domain components or using components from one specific domain in another domain.

It is feasible to reuse components together with evidence and/or safety arguments by validation and verification activities and demonstration of conformity to currently used safety standards. Anecdotal experience in the railway domain indicates that it is feasible to reuse components and safety arguments only when a component is a standalone generic product with its own full lifecycle and certification. In this domain, reuse of safety arguments (existing safety case and assessment) is sometimes possible without.

However, full reuse may fail when the demands of national regulatory or safety standards vary from product development demands. In this case additional assurance activities are required to bring up the additional evidence to satisfy national regulations; in some cases, modifications to the design or usage conditions may also be necessary.

Generally, engineering reuse of “smaller” components (e.g. software or hardware modules within generic products) does not lead to significant reuse of any associated elements of the safety demonstration. These have to be reworked and/or reassessed. Exceptions exist for some discrete and bounded items such as a communications protocol offering particular safety-related properties.

---

1 This degree of rigour is sometimes expressed as a discrete level – such as Development Assurance Level (DAL) or Safety Integrity Level (SIL).
3 Safety certification

This section of the report provides an overview of the notions of safety certification in general and modular certification in particular. From the onset it is important to stress that, within OPENCOSS, we use the term “certification” broadly to denote verifiable demonstration of safety of a particular product (as described in the introduction), in compliance with a given standard or norm. This typically includes demonstration that safety risks associated with a safety-critical system or equipment (within a particular usage context) are reduced to acceptable levels whilst also demonstrating conformance with the requirements posed by the relevant standards. There are a number of reasons why such ‘certification data’ (or parts thereof) may not be submitted to a formal certification process:

- Some of the safety-critical industries – for instance automotive – do not have legal regulation or strong quasi-regulatory stakeholders (such as Independent Safety Assessors).
- In some tightly regulated jurisdictions (e.g. Civil Aviation), only certain products (e.g. aircraft and engines) are certified. Whilst the developers of lower-level products (e.g. individual aircraft systems or equipment) need to demonstrate achievement of certain safety requirements and compliance with relevant standards, limited formal recognition is given for certification data related to their components.

3.1 Safety Certification approaches

There are a number of different approaches taken to system safety management and certification, which differ both across and within domains. Some domains have certification authorities, which provide legal approval for a product or system before it can be used, for example the aviation and railway domains. Other domains, such as automotive, have no official certification processes, but do have sets of industry-developed safety standards, which are used to help ensure that levels of safety are maintained.

Overall, all of the industrial domains considered within the OPENCOSS project (primarily automotive, aviation/avionics and railway) have some safety standards, see Table 1. It is important to note, that whilst the standards may vary in the degree of prescription, their requirements almost invariably require interpretation before conformance can be demonstrated. In regulated jurisdictions, this interpretation (and, where applicable, deviation from the standard practice) requires explicit documentation and is subject to negotiations with the authorities2. This means that compliance demonstration even within the jurisdictions of relatively prescriptive standards is not entirely objective and judgement-free. The subjective elements of certification are likely to pose some key challenges to compositional or cross-domain certification.

---

2 For example, whilst the Flight Control System of Falcon 7x aircraft has been certified on the basis of relatively novel AltaRica models (rather than classical fault trees), the certification authority (EASA) has imposed additional requirements on demonstration of conformance of model management and analysis tools (Cecilia OCAS and ARBOR) with DO-178b standard (that is normally considered not applicable to RAMS analysis software).
Baseline for the compositional certification approach of the OPENCROSS platform

<table>
<thead>
<tr>
<th>Industrial Domain / Sub-domain</th>
<th>Applicable Standard Reference</th>
<th>Standard Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automotive</td>
<td>ISO 26262</td>
<td>Road vehicles – Functional safety</td>
</tr>
<tr>
<td>Aviation</td>
<td>FAR/CS25 - §1309</td>
<td>Certification Specification for Large Aeroplanes, Equipment, Systems and Installations</td>
</tr>
<tr>
<td></td>
<td>AMC 25.1309</td>
<td>Acceptable Means of Compliance – System Design and Analysis</td>
</tr>
<tr>
<td>Systems</td>
<td>ARP4754a</td>
<td>Certification Considerations for Highly-Integrated or Complex Aircraft Systems</td>
</tr>
<tr>
<td></td>
<td>ARP4761</td>
<td>Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment</td>
</tr>
<tr>
<td>Software</td>
<td>DO-178c</td>
<td>Software Considerations in Airborne Systems and Equipment Certification</td>
</tr>
<tr>
<td>Avionics</td>
<td>DO-297</td>
<td>IMA Development Guidance and Certification Considerations</td>
</tr>
<tr>
<td>Rail</td>
<td>EN 50129</td>
<td>Railway Applications – Communications, signalling and processing systems – Safety related electronic systems for signalling</td>
</tr>
<tr>
<td></td>
<td>EN 50126</td>
<td>Railway Applications – The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS)</td>
</tr>
<tr>
<td>Software</td>
<td>EN 50128</td>
<td>Railway Applications – Communications, signalling and processing systems – Software for railway control and protection systems</td>
</tr>
<tr>
<td>Cross-Domain</td>
<td>IEC 61508</td>
<td>Functional safety of electrical/electronic/programmable electronic safety-related systems</td>
</tr>
</tbody>
</table>

Table 1: Key Safety Standards in Automotive, Civil Aviation and Railway domains

Some safety documents are described as standards, e.g. IEC 61508 in several domains, whereas others are described as guidance, such as DO-178B in the avionics domain. In practice, those documents described as guidance are almost always used in their application domain, and deviation from them can be difficult to justify. For the sake of simplicity, the term ‘safety standard’ is used here to denote both kinds of document.

The safety standards describe generalised approaches to identifying hazards and risks, design lifecycles, and analysis and design techniques. By using the appropriate domain standards, a system developer can develop their system in a way that allows them to achieve certification or approval. However, as noted above, compliance to a particular standard may not be a black and white issue, there can be some degree of choice as to which techniques should be used, and it may not always be clear what depth to apply them to.

One of the key aims of OPENCROSS is to support cross-domain certification; therefore it is crucial that we understand the different certification approaches and standards from each domain. The following list attempts to qualify the types:
• Process-based certification (see also Chapter 4 in OPENCOSS D4.1) - the emphasis in standards and certification is on following a prescriptive design process, with lists of required or expected analysis techniques given for each development stage. The depth and extent to which these are applied typically depends on an early analysis of the "integrity" of the system, its components or functions. Care is needed here, as integrity levels in one domain may not be directly analogous to assurance levels in another.

• Prescriptive product-based certification - the emphasis is on the product that is being developed, and demonstrating that specific (low-level) design rules have been followed or that the system includes particular design features or functionality.

• Objective-driven product certification - This approach requires the use of a safety case consisting of a safety argument and backing evidence. The safety argument should contain a reasoned argument as to why the system is acceptably safe to operate in a particular context. The supporting evidence would include the sort of analysis and testing described in more prescriptive standards. In practice, this approach includes both product- and process-related information. It may also include a compliance argument, relating to how well a standard has been met (see also OPENCOSS D4.1).

In practice, many industrial sectors and regulatory jurisdictions combine different certification approaches.

3.2 Modularity and compositionality in system design

Whilst WP5 focuses on compositional certification – rather than design or analysis – the notion of modularity in system design is an important part of the context of work of the work package.

In order to cope with the complexity of hardware and software, we use architecture and design techniques that use the "divide and conquer" idea to cut up the problem into sub problems of manageable proportion. In software and software engineering this is called modularization (Parnas, 1979) (Stevens, Myers, & Constantine, 1974) (Beck & Diehl, 2001). Splitting problems up gives us clear advantages. For example, each sub-problem can be addressed, understood, and implemented (in a component) individually. Components or modules can later be maintained in isolation (i.e. with minimal impact on other components or modules).

Depending on the operational context, modules and components may be different things, but for now, we consider both modules and components to describe the same thing: a grouping of functionality. We will use components henceforth. Components, being groupings of (similar) functionality, can communicate with other components to get the responsibilities done.

Measurements on the modularization can give us insight into the strengths and weaknesses of a system design. Ideally, the design should be “loosely coupled” (or low coupling) and have “high cohesion”. Loosely coupled means that any component does not depend on too many other components. High cohesion means that elements of this component are strongly related (e.g. things that belong together, are placed together). This way, maintenance of a system can be made minimally intrusive because of the fact that related things are in the same place and not too many other components have something to do with it.
Examples or such measurements are “fan-in”, measuring the number of components that use our component and “fan-out” measuring how many other components ours uses.

Hiding the internal design and design decisions in a component from other components is referred to as “information hiding”. Information hiding improves the modularity of a design. Because other components do not see the internal structure and design decisions within a certain component, they cannot use this information or depend on it. It can only communicate through interfaces, creating a barrier that allows locally focused maintenance without having to address all dependent components as well.

Splitting up a software problem into components is not an exact science. It requires an architect or software designer to decompose the problem to its essential building blocks, grouping these blocks based on similarity into components and designing inter-component-communication in such a way that the result is “loosely coupled” and has “high cohesion”. The main rule of thumb is called “separation of concerns” (Dijkstra, 1982) in which one tries to separate the system into distinct features that have minimal overlap in functionality and responsibilities. More in-depth information can be found in Section 6.

Whilst the notions of compositional design and compositional certification are distinct they are not entirely independent. Firstly, as a relatively immature topic, compositional certification is largely based on the same abstract principles as compositional design: design-and-conquer, information hiding, low coupling and high cohesion. The difference is that in compositional certification those principles are applied to the certification artefacts (i.e. safety cases) rather than to the systems themselves or their various models.

Both compositional certification and design rely on the notion of [system or safety case respectively] an “architecture” that guides how individual “building blocks” are interrelated and how they can be combined into a coherent and complete “whole”. The structure of system and safety case architectures are not necessarily related – i.e. safety case modules do not have to always relate to components within the system. However, in many cases alignment between the two architectures can be beneficial. In fact business case for compositional certification (Fenn, Hawkins, Williams, Kelly, Banners, & Oakshott, 2007) often relies on a degree of alignment between the two architectures and, consequently, between the organisation of the safety case into modules and the responsibilities of the supply chain stakeholders. Some system architectures – such as IMA (see section 4.2.1) – may facilitate compositional certification more than others.
4 Purpose of safety standards from reuse perspective

The intent of system safety standards is to ensure that safety critical systems are developed in such a way that they never pose more risk of harm than is deemed acceptable. A safety-critical system is one whose failure may cause death or injury to people, or harm to the environment. It also implies that such systems are trustworthy. This has not only an impact on the functioning of the system, but also on its reliability. In essence safety requirements make good system engineering an implicit precondition of developing safety-critical systems.

System safety engineering and assurance in all three industrial domains directly represented within OPENCOSS consortium (automotive, aviation/avionics and railway) is standardised. The level of prescription and detail of the standards varies both within the sectors (e.g. between different standards in aviation domain) and across the domains boundaries. In more regulated domains standards and guidance documents can be often organised into a hierarchy where top-level standards (e.g., CS/FAR25 in aviation) set general safety objectives, whereas lower-level norms either define standard frameworks for achieving those targets (e.g., ARP4754/4761 and DO-297 documents in aviation and EN50129 in railway domain) or refine them into relatively detailed process or product requirements (e.g., DO-198b/c and EN50128 in aviation and railway respectively). It is not unusual for the necessary degree of conformance with the standard (from the perspective of the regulator or accepted industrial practice) to decrease as the level of detail and prescription of the standard requirements increases.

However, it is important to note that requirements of almost all modern safety standards require (and permit) significant degree of interpretation and variation. This means that safety certification (or, more broadly, safety assurance) can never be reduced to a purely prescribed activity and is always in part based on a subjective engineering judgement. This subjectivity and interpretation pose one of the key challenges to the compositional certification studied within WP5 of OPENCOSS, especially with respect to composition and reuse of certification artefacts across the boundaries of engineering organisations, regulatory jurisdictions, and communities of practice.

This section introduces the key standards of different industrial domains, focusing on those of the domains addressed in OPENCOSS. The section also discusses their general philosophies and frameworks. The authors pay particular attention to the aspects of the standards related to compositional assurance/certification. Indeed, the notions of Safety Element out of Context (SEooC) and Generic Product Safety Case promoted by ISO26262 and EN50129 standards in the automotive and railway domains respectively embed the concept of compositional certification. Although within the aviation domain the concept is not explicitly recognised by the standards, the framework of ARP4754 and ARP4761 documents implies that some of the aircraft safety assurance can be undertaken in relative independence (i.e. compositionally) for individual aircraft systems. The concept of compositional assurance is further reinforced for Aircraft IMA by the existence of DO-297 guidance.

Whilst in this document we focus on the modularity of certification artefacts with respect to the structure of the system design, it is also important to note that this is not the only approach to decomposing of certification activities and work products. Examples of other not mutually exclusive strategies include:
• Decomposition by the types of activities identified within the standards’ safety lifecycles (e.g. Functional Hazard Assessment, [Preliminary] System Safety Assessment and various Common Cause Analyses in ARPs 4754 and 4761)
• The refinement/instantiation concept implicit in the relationships between Generic Product, Generic Application and Specific Application Safety Cases described in EN50129
• Standard recommended organisation of the safety cases into reports (and further organisation of the Technical Safety Report into specific sections) prescribed by EN50129

The rest of this chapter consists of four subsections. Section 4.1 focuses on the automotive domain, Section 4.2 for avionics domain, and Section 4.3 for the railway domain. Finally, Section 4.4 widens the scope and takes a look at relevant standards from other domains.

4.1 Automotive

As was mentioned above, within the automotive domain the current safety standard is a recently published ISO 26262. Within this, the notion of composability of the conformity process is supported by three key concepts:

(1) The “Proven-in-use argument” process,
(2) The tailoring of the safety activities in the lifecycle and
(3) The SEooC concept.

These first two concepts above are also are supported by the traceability requirement. This acts as a transversal control feedback that applies at the various levels of the lifecycle.

The “Proven-in-use argument” is an alternative means of compliance with ISO 26262 that may be used in the case of reuse of items or elements when their field data is considered to provide adequate evidence that the item or element is sufficiently trustworthy. A proven-in-use argument can be applied to any type of item/element whose definition and conditions of use are identical to - or have a very high degree of commonality with - a product that is already released and in operation (ISO 26262 – Part 8, Clause 14). Proven-in-use argument can be used for a range of items such as a system, function, hardware or software component (each such item is referred to as “candidate”).

To substantiate proven-in-use arguments, ISO 26262 requires the following supporting information to be provided:

• Regarding the intended use of a candidate component:
  o candidate item specification
  o applicable safety goal(s) or safety requirement(s) with corresponding ASIL(s)
  o foreseeable operational situation and intended operating modes and interfaces.

• Regarding the previous use of existing item or element for a candidate:
  o field data from the service period (from an external source)
  o available (parts of a) safety case.
The process for the candidate item collects the results of a set of safety lifecycle (sub-) phases based on the following:

- Description of the candidate item and its previous use (interface and environmental, physical and dimensional, functional and performance characteristics, functional and safety requirements)
- Trace of the changes on the candidate item and its environment
- Field data analysis (configuration management, change management, target values, field problems/incidents).

The results of the above analysis shall be introduced in the safety plan. The plan also includes a report containing the analysis of changes to the candidate item and the analysis of field data. Subsequently, an external and ‘formal’ confirmation review of the proven-in-use argument of candidate item shall be performed to evaluate the results of the proven-in-use analysis.

The second key compositionality concept - the “tailoring” - is focused on the simplification of the safety activities for the development of a specific item. In simple terms, this concept allows to omit or perform in a different manner (than that described in ISO 26262 standard) an activity of the safety lifecycle. This is, however, subject to satisfaction of a number of requirements stipulated by the standard (see ISO 26262, Part 2, Clause 6.4.5). The justification of this ‘deviation’ must be provided, together with the corresponding version of the safety plan.

It should be noted that utilisation of the proven-in-use argument by definition requires tailoring (although, of course, tailoring may be used for other reasons too). This is because, when proven-in-use argument is used, some of the stages of the standard safety lifecycle for the candidate item are omitted (with pre-existing item’s data being reused instead of being re-generated).

The final compositionality concept of ISO 26262 is the Safety Element out of Context (SEooC). In essence this is mainly a reuse concept that promotes modular designs. A typical example is a component (or a subsystem) being reused in several higher-level systems.

In some respects, SEooC concept is based on similar principles to tailoring of the proven-in-use argument. The main difference is that SEooC is related to a new development of a reusable component rather than to a reuse of an already existing, well-known and validated product. SEooC is a general and re-usable component developed for some foreseeable hypothetical application. This new component can be re-used in a variety of (different) contexts, subject to provision of the required justification and validation as well the appropriate revision of the safety plan accordingly. When developing or reusing a SEooC, some of the safety lifecycle activities are tailored (ISO 26262, Part 2, Clause 6.4.5.6) to avoid unnecessary replication of the activities.

The specific characteristic of the SEooC concept is that it is a safety-related element that is not developed for a specific item. The SEooC can be a system, a set of systems, a subsystem, a software component, a hardware component or a part of a hardware component. The development of a SEooC involves making assumptions on requirements of its corresponding phase in ISO 26262 safety lifecycle, in relation to the actual component. These assumptions are related to its use and context, but also to the external interfaces Figure 4. These assumptions are verified during integration into the actual item.
The SEooC is developed considering assumptions on an intended functionality in a certain context. These assumptions are set up in a way that addresses a superset of items, making it possible for the SEooC to be used in multiple distinct (but sufficiently similar) items later. The developer of the SEooC shall provide the assumed requirements and assumptions related to the wider design where the SEooC will be used and integrated.

The integrator that uses a SEooC (e.g. a vehicle manufacturer) is responsible for:

- Establishing the validity of the assumptions used (and documented) during SEooC development with respect to the (new) instantiation/use context (e.g., assumptions on the overall design of the system where it is proposed that the SEooC should be used);
- Verification activities to demonstrate that the developed SEooC is consistent (at any level) with the requirements in the context where it is intended to be used. The functional safety requirements of the item are matched with the functional safety requirements assumed for the SEooC, to establish their validity.

The development of a SEooC can start at a certain hierarchical-level of requirements, and design of the actual item, according to the tailoring requirement as previously stated.

As was mentioned in the beginning of this section, the traceability requirement applies to the processes associated with the proven-in-use argument process, the tailoring of the safety lifecycle activities and the SEooC concept. Overall, the traceability is a necessary condition in the documentation of the entire safety lifecycle and assessment. Moreover, it becomes particularly important when dealing with composability, since the changes, modifications, variations of the item and of the applied safety lifecycle steps must be appropriately identified, documented and made available for the review. From the business drivers perspective the traceability is also fundamental for enabling a lean validation/assessment workflow and, thus, maximising the benefits of compositional working.
4.2 Avionics/aviation

The DO-178B/ED-12B document (Software Consideration in Airborne Systems and Equipment Certification) deals with safety of software that is used in airborne systems. Although not mandatory, not using DO-178B/ED-12B (hereafter described as DO-178B) is extremely difficult to justify. It has been created as a joint effort of the Radio Technical Commission for Aeronautics (RTCA), a US non-profit organization that develops technical guidance for use by government regulatory authorities and by industry, and The European Organisation for Civil Aviation Equipment (EUROCAE), a non-profit organization for avionics stakeholders such as Manufacturers, National and International Aviation Authorities and users and Service Providers. DO-178B is applied by the Federal Aviation Administration (FAA), the United States national aviation authority.

DO-178B represents industry consensus (both industry practitioners and certification authorities) opinion on the best way to ensure safe software in airborne systems and equipment (Ferrell & Ferell, 2001). Its application results in a level of assurance (from A to E) in the correct functioning of the software in compliance with airworthiness requirements. While DO-178B contains objectives for the entire software development life cycle, there is a clear focus on verification. DO178B defines objectives for lifecycle processes of planning processes, development processes (requirements, design, code and integration), and the integral processes (verification, configuration management, software quality assurance, and certification liaison).

It is important to note that software is never certified as a standalone entity. Application of DO178B fits into a larger system of established or developing industry practices for systems development and hardware. The system level standard, that addresses the total life cycle of systems that implement aircraft level functions, is SAE ARP4754 (Certification Considerations for Highly-integrated or Complex Aircraft Systems), to be assessed by SAE ARP4761 (Figure 5. An equivalents standard to DO178B exists for hardware development in the RTCA/EUROCAE DO-254 ED-80 document, for process and flow of information between hardware processes and system process. DO178B specifies the information flow between system processes and software processes (in order to keep track of requirements allocated to software). For system integration and test, the RTCA DO-160D Environmental Test Specifications are to be used.
When software is certified to DO-178B standard, the underlying safety argument and its claims are largely implicit (Rushby, 2007). New business practices, such as using Commercial off the Shelf (COTS) ready-made products, outsourcing, and continuous evolution ask for new ways of certification: a goal-based approach. Compositional certification can be classified as a goal-based approach, certifying various components that are running separately, but also in cooperation within a platform. Compositional certification can lead to lower costs of developing and maintaining systems by reusing components in other systems, and by lowering recertification costs due to prediction of target areas where changes (new or modified components) will take place. The challenge for compositional certification is that in order to certify components in complex (sub)systems, one should fully understand the relations and dependencies of all of the different components in a certain system-context.

RTCA DO-297 is a relatively new standard (December 2005) that supports the development and assurance of Integrated Modular Avionics (IMA). IMA is the term used for a distributed computing network aboard aircraft, which supports avionics software of many different assurance levels, and is designed for flexibility in configurations and modularity. As a complement to DO-178B, this standard goes beyond software development guidelines. Although not mandatory, DO-297 addresses the system level considerations for incorporating modularized avionics systems within an ideally seamlessly integrated holistic aircraft. It supports simpler maintenance of applications (reducing the amount of re-verification effort needed following a change) to lower overall system through-life lifecycle costs. It also supports software evidence reuse to reduce effort required when re-using components in different systems.

IMA qualification and assurance (Eveleens, 2006) is concerned with the certification authority acknowledging that the module, application, or system complies with its defined requirements. Acceptance is recognition by the certification authority (typically in the form of a letter or stamped data sheet) signifying that the submission of data, justification, or claim of equivalence satisfies applicable guidance or requirements. The goal of acceptance is to achieve credit for future use in a certification project (initial credit must be gained within an actual project and cannot be sought independently of this). IMA
technology has introduced the possibility to fragment the certification process into several tasks by incremental acceptance. The DO-297 standard describes the certification tasks as shown in Figure 6 as:

1. Module and/or platform acceptance;
2. Application acceptance (soft- and hardware);
3. IMA system acceptance (integration of multiple applications);
4. Aircraft integration;
5. Change of modules or applications and;
6. Reuse of modules or applications.

![Figure 6: IMA Development Guidance and Certification Considerations (Eveleens, 2006)](image)

### 4.2.1 Civil Aviation ARP4754 & ARP4761

Whilst formally only guidance on the acceptable means of compliance with the Certification Specification FAR/CS25, these two documents are accepted by all major aviation regulators and airframers as de-facto standards for the safety assessment of aircraft and its systems. The documents describe the overall process that should be followed for assessment of the aircraft (safety lifecycle) and how this should be integrated within the overall development process. The guidance provided by Aerospace Recommended Practices (ARPs) is technology-independent and covers system architecture, design, and integration and verification stages of the development lifecycle. The standards do not provide specific requirements for software or any guidance for detailed design and implementation of items. These areas are covered by other international standards such as DO-178B, see Figure 7.
ARPs define a number of activities that should be carried out at the level of the aircraft as a whole as well as at the level of individual aircraft systems (and sub-systems). This covers both design-time and requirements-driving activities as well as the confirmatory V&V activities. The design-driving activities consist of aircraft- and system- level Functional Hazard Assessment (FHA) and Preliminary System Safety Assessment (PSSA). The former is used to identify hazards (in ARPs’ terminology – Failure Conditions), determine their severity classification and establish top-level safety requirements, whereas the latter is used to systematically assess feasibility of the design proposals and to iteratively decompose both quantitative and qualitative safety requirements and allocate them to individual items. The confirmatory activities are mostly based on the System Safety Analysis (SSA) that gathers and combines items characteristics to verify whether aircraft- and system- level requirements have been met. ARPs are not prescriptive in the sense that they do not mandate which techniques should be used for fulfilling the objectives of each stage of the safety lifecycles. In fact, the documents explicitly state that there are a number of alternative techniques that may be used at every lifecycle stage. For PSSA and SSA stages, the documents provide an illustration of use of three techniques: Fault Tree Analysis, Reliability Block Diagrams (also known as Dependence Diagrams), and Markov Models/Analysis.

The core safety lifecycle activities (FHA, PSSA and SSA) are supported by the Common Cause Analysis (CCA), whose aim is to identify covert dependencies between otherwise seemingly independent systems (and items) based on, among other considerations, similarity in their design/technology, spatial relationships, commonalities in maintenance procedures, and vulnerability to particular physical threats (such as tyre burst, engine disc failure and lightning). The implicit goal of CCA is to compensate for the decomposition
nature of the core safety activities and their inherent vulnerability to unverified independence assumptions.

With exception of Aircraft Level FHA and CCA, activities advocated by ARPs are carried out at the level of individual aircraft systems, which implies some degree of compositional certification. Furthermore, the milestones of the aircraft type certification process and review meetings with the regulator are typically organised around the stages of the ARPs safety lifecycle and – for PSSA and SSA – aircraft systems. This somewhat reinforces the notion of compositional certification on the one hand whilst also highlighting another form of modularity in certification data on the other (i.e. with respect to lifecycle phase and activities).

However, because of the high degree of coupling between the systems introduced by the shared resource management systems (Hydraulic and Electrical Power Distribution and IMA), recent revision of the documents (ARP4754a and ARP4761a) have strengthened the non-compositional safety assessment activities by introducing two new stages into the safety lifecycle: Preliminary Aircraft Safety Assessment (PASA) and Aircraft Safety Assessment (ASA). These can be seen as multi-system PSSA and SSA respectively.

Finally, it is important to state that, whilst the civil aviation industry has been historically reluctant to adopt the notion of the “safety case”, this has now been explicitly introduced in the revision of ARP4754a. The document mandates that a safety case must “communicate a clear, comprehensive and defensible argument that the aircraft and systems are acceptably safe to operate in a particular context”.

4.3 Railway

CENELEC standards provide Railway authorities and the railway industry with a process that enables the implementation of a consistent approach for the management of reliability, availability, maintainability and safety (RAMS hereinafter). In more detail, the standards that describe this approach are EN50126, EN50128, and EN50129.

These three norms are applicable to the specification and demonstration of RAMS for all railway applications/system and sub-systems, including those containing software, and in particular according to paragraph 1.2 of EN50126:
- to new systems;
- to new systems integrated into existing systems in operation prior to the creation of these;
- to modifications of existing systems in operation prior to the creation of this standard;
- at all relevant phases of the lifecycle of an application.

EN50126 (paragraph 1.1 of EN50126):
- defines reliability, availability, maintainability and safety and their interaction;
- defines a process for managing RAMS, based on the system lifecycle;
- enables conflicts between RAMS elements to be controlled and managed effectively;
- defines a process for specifying requirements for RAMS and demonstrating that these are achieved;
EN50128 describes methods, tools and techniques to develop software systems that meet the demands for safety integrity placed upon them in accordance to process defined in EN50126.

Finally, EN50129 defines a process for the acceptance and approval of electronic railway signaling systems. This standard is concerned with the evidence to be presented for the acceptance of safety-related systems. It specifies the lifecycle activities which shall be completed before the acceptance stage, followed by planned activities to be carried out after the acceptance stage. This evidence shall be presented in a structured way. For this reason EN50129 suggests the use of the Safety Case, a document in which all evidence of lifecycle activities shall be collected and organized. EN50129 groups these evidences in three main categories:

- evidence of quality management
- evidence of safety management
- evidence of functional and technical safety

All evidence shall demonstrate that conditions necessary to accept a safety-related system as adequately safe are satisfied at equipment/system/sub-system levels. The Safety Case shall be submitted to the Safety Authority in order to obtain safety approval for the system under assessment.

EN50129 reports the structures that the Safety Case shall present (from paragraph 5.1 of EN50129):

- Part 1 – definition of system/equipment: definition or reference of the system/equipment to which the Safety Case refers, including version numbers and modification status of all requirements, designing and application documentation.
- Part 5 – related safety cases: references to the Safety Case of any sub-system or equipment on which the main Safety Case depends. It shall also demonstrate that all the safety-related application conditions specified in each of the related sub-system/equipment Safety Cases are either fulfilled in the main Safety Case, or carried forward into the safety-related application conditions of the main Safety Case.
- Part 6 – conclusion: summarize the evidence presented in previous and argue that the relevant system/sub-system/equipment is adequately safe, subject to compliance with the specified application conditions.

Paragraph 5.5.1 of EN50129 describes three different categories of Safety Case:

- Generic Product Safety Case (independent of application): a generic product can be reused for different independent applications
- Generic Application Safety Case (for a class of application): a generic application can be reused for a class/type of application with common functions;
- Specific Application Safety Case (for a specific application): a specific application is used for only one particular installation.

From a safety perspective it is essential to demonstrate that generic application conditions are satisfied by specific application taken into account for safety assessment. The structure of Safety Case and Safety Approval procedure of the three categories described above are basically the same.
Only for specific applications some additional work may be required since for their safety approval it is necessary to separate their physical implementation from application design of the system. For this reason EN50129 recommends to divide safety case into two portions for specific applications, (from paragraph 5.5.1):

- The Application Design Safety Case: this shall contain the safety evidence for the theoretical design of the specific application.
- The physical implementation Safety Case: this shall contain the safety evidence for the physical implementation of the specific application.

### 4.3.1 Safety Approval Process

For an application to be safety assessed by a Safety Authority, first a Safety Assessment Report shall be obtained from an Independent Safety Assessor (paragraph 5.5.2 of EN50129). This report shall provide additional assurance that the railway application under assessment has achieved the requested level of safety.

The report should explain the activities that the Safety Assessor has carried out to determine how the system or item of equipment (hardware and software), has been designed to meet its requirements, and possibly specify some additional conditions for the operation of the system/sub-system/item of equipment. These additional conditions imposed by the Safety Assessor shall be fulfilled to obtain Safety Approval by a Safety Authority.

Once it is demonstrated (within the Safety Case) that all safety conditions have been satisfied) the system may be granted safety approval by the relevant Safety Authority. It is important to underline that, for a generic product, and for a generic application, it should be possible for safety approval granted by one Safety Authority to be accepted by other Safety Authorities (i.e., cross-acceptance). This is not considered possible for specific applications. The safety approval process, for all three categories of a Safety Case, is illustrated in Figure 8.
Figure 8: Safety Acceptance and approval process taken from EN50129
4.3.2 Dependency between Safety Approvals

As described at the end of the previous section, if the Safety Case for a system depends on the Safety Cases of other sub-systems or equipment, then Safety Approval of such kind of system is not possible without previous Safety Approval of the related sub-systems/equipment, according to paragraph 5.5.4 of EN50129. If Safety Approval has been obtained for a generic product, or for a generic application, a reference may be made to this in the application for Safety Approval of a specific application. It is not necessary to repeat the generic approval process for each application. This dependency between Safety Approvals is illustrated in Figure 9.

In this situation, Safety-Related Application Conditions stated in the Technical Safety Report of each Safety Case shall be fulfilled in the higher level Safety Case, or are transferred into the Safety Related Application Conditions of the higher level Safety Case.

![Diagram of dependencies between Safety Cases / Safety Approval]

**Figure 9: Examples of dependencies between Safety Cases / Safety Approval**

4.4 Other domains: Defence

Although not directly in scope of the OPENCOSS project, UK defence standard 00-56 (DSTAN 00-56) issue 4 contains relevant information for military certification. Its scope, includes vehicles and aircraft, thus this domain may provide one area of potential cross domain re-use of components. It was released in 2007 replacing the heavily process based standards that used to be used in the domain. DSTAN 00-56 is a goal-based standard, which emphasises a risk management process, and it is prescriptive only about the need to produce detailed risk and safety management plans, and a safety case showing that the risks inherent in the system have been sufficiently mitigated and managed. The standard states that:
"The Safety Case shall consist of a structured argument, supported by a body of evidence that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given environment."

The standard doesn't prescribe how the evidence is to be gathered, or which techniques are to be used, instead it offers guidance encouraging the use of existing civil standards, e.g. DO-178B for software, as appropriate. However, the evidence must be presented in a way that makes it clear how safety requirements have been managed, i.e. as part of a bespoke safety case.
5  Overview of the state of industrial practice

The previous section has presented and discussed requirement of safety standards that are related to compositional certification and reuse of certification artefacts. We have demonstrated that standards applicable to all three industrial domains directly considered by the project do acknowledge the principles of compositional certification. However, standards themselves do not constitute industrial practice. They merely constrain practices and methods that can be adopted by the company. These constraints are often expressed in relatively general terms that require interpretation by the regulated companies; standards also permit considerable degree of variation in adopted practices.

This section of the report describes how the target domain industries conform to, and follow, safety standards (with particular focus on standards’ requirements related to compositional certification). I.e., it looks at their practical rather than theoretical application. The industrial domains organise and manage the certification and conformity assessments through different approaches in relation to their own products. The description contained within this section (including examples) is based on the extensive relevant experience of all WP5 partners both first hand and obtained through supply-chain and consultancy collaboration.

5.1  Automotive

Within the automotive domain there is currently no consistent ‘standard’ practice with respect to compositional certification. The current safety standard of the domain – ISO26262 – has only been recently issued. At the time of writing of the present report, insufficient amount of time has passed for the industry to produce an interpretation of standard’s requirements and to converge on the set of practices.

However, many of the industrial needs that underlie compositional certification are present within this domain. In particular there is a high level of component reuse across different vehicles and development of key safety critical components is routinely undertaken by suppliers rather than vehicle integrators. This is particularly the case for the various electric and electronic elements for engine control, vehicle handling, energy management, dashboard communication, and internal comfort functionality.

The process is currently managed by vehicle integrators on an organisation-by-organisation basis. Vehicle integrators specify requirements for components (including safety requirements) and standards that suppliers have to comply with. Those standards are typically developed by the integrators themselves although they usually are based on- and sometimes cite- various ‘external’ international safety standards (such as IEC61508). The acceptance process is not yet structured and is based on the evidence of the requirements fulfilment verified by testing at supplier premises and by the automaker. The conformity assessment is performed on the basis of all the documentation required by the internal and international

---

3 Because of the absence of external regulation with respect to system safety automotive domain does not have a notion of “certification”; the term “conformity” (and “conformity assessment”) is used to refer to the degree to which a vehicle satisfies external and internal safety standards. Within this document, however, we use the term “certification” for consistency.
standards, related to the application under analysis and required in the project plan. Overall this process is sometimes iterative and not always satisfactory in terms of cost and time to market boundaries.

The standards that were used within the automotive industry – prior to ISO26262 publication – have not covered relationships between vehicle manufacturers and component suppliers and, thus, have not facilitated convergence of industrial practices with respect to compositional certification. Based on authors discussions with OPENCOSS partners and within our wider networks of collaborators it became apparent that many industrial stakeholders in this domain are hopeful that publication of ISO26262 will result in higher-level of harmonisation and will lead to more effective collaboration within the supply chain. It is our belief that OPENCOSS project in general and WP5 in particular have an opportunity for achieving high industrial impact in the automotive domain.

5.2 Avionics/aviation

Avionics is the generic term used for electronic sub-systems on airborne vehicles, i.e. aeroplanes, helicopters, satellites, spacecraft etc.. There are different sets of standards for each set or type of vehicle, and also for different domains. For example, military aircraft must comply with military standards, but also some civil standards if they are to be used in civil airspace (e.g. so that they can be landed at civilian airports). Obviously, there are some similarities in the way that all airborne vehicles are deployed and they may share certain hazards (e.g. controlled flight into terrain), but the ways in which this can occur (and the severity of consequences) may vary wildly. This document concentrates on civil avionics as the largest and most well specified market and deployment arena.

5.2.1 Civil aviation

As discussed in Section 4.2.1, within the civil aviation sector, only two types of products undergo a formal certification process – aircraft and engines. In this document we focus on aircraft certification (typically referred to a “type certification”) and on the certification activities related to safety of the aircraft.

Based on the experience of the authors, the format of certification artefacts varies across airframe/aircraft integrators considerably. Currently, it is largely agreed at the early stages of the development between the airframer and the certification authority (e.g. FAA or EASA) on a case by case basis. This situation can be partially explained by a relatively small number of companies that develop whole aircraft in a jurisdiction of each regulator. It should be also noted that, until recently, de-facto standards in this sector (ARP4754 and ARP4761 documents) paid little attention to the certification artefacts and safety cases. This situation is changing in the recent revision of the documents; however, those standards enhancements have not yet been reflected in the industrial practice.

Consequently, currently, the safety certification practices within the civil aviation sector are structured around key analysis activities, following a “safety lifecycle” prescribed by the ARP4754 and ARP4761 documents. Whilst formally only the final design of the aircraft is certified, in practice the certification process starts informally from the earliest stages of development with key negotiation milestones between airframers and regulators, generally reflecting the core stages of the safety lifecycle of the ARP documents: Aircraft FHA, Systems FHA(s), PSSA(s), and SSA(s).
In the simplified view of the process presented in Figure 10:

- Aircraft and System FHAs derive top-level safety requirements for the aircraft and its individual systems (respectively);
- PSSA verifies that proposed systems designs are feasible in the light of those requirements, iteratively decomposes top-level safety requirements into derived safety requirements (DSRs) and allocates those to the items, and;
- SSA verifies that systems as designed and implemented meet requirements as identified in FHA.

This simplified view of the assessment process is strictly ‘decompositional’, and requirements are iteratively decomposed and allocated in a top-down manner with no cross-dependencies between the systems.

In the real development process, however, because of the dependencies between different aircraft systems, safety requirements for a given system can be identified in the course of the assessment (e.g. PSSA) of another aircraft system. The most obvious examples are dependencies between resource management systems (IMA, Hydraulic and Electrical power generation and distribution) and consumer systems. For instance, the aircraft wheel breaking system (WBS) and the landing gear extension/retraction system (LGERS) will depend on shared power and IMA. In other words, the likelihood of WBS or LGERS failure conditions (hazards) will depend on the likelihood of various failure conditions of the resource systems. Similarly, as safety requirements for WBS or LGERS are decomposed and allocated, some of the generated DSRs will relate to IMA, Electrical or Hydraulic Systems. These DSRs are not necessarily aligned with the safety requirements for resource management systems identified in the FHA. They may impose more limiting probabilistic budgets on failure conditions (than those identified in the FHA) or may even identify new failure conditions (or, most likely, new combinations of failure conditions that were not explicitly considered in FHA).

---

4 For example, analysis of the WBS may identify some inability on the electrical power distribution system to provide power on a particular combination of busbars as being of particular concern from the WBS perspective. At the same time, it is unlikely that the electrical system FHA would explicitly consider every combination of busbar ‘faults’ as a unique system-level failure condition (and, thus, it is unlikely that safety requirements will be generated for each such combination during the FHA).
Traditionally, these cross-dependencies between the systems and the ‘lateral’ flow of DSRs across systems boundaries have been managed through the notion of “dependent system failures” (DSFs). It is important to note that even this simple concept poses challenges to the compositional certification of aircraft systems since it establishes contextual dependencies between certification data associated with different systems. Furthermore, the allocation of requirements related to DSFs typically relies on significant and relatively informal multi-party negotiations between airframers and (numerous) system integrators.

Most importantly, with increasing integration between aircraft systems the cross dependencies between aircraft systems are becoming too complex to be managed through requirement traceability tools. On a typical modern aircraft such dependencies may form complex non-immediate loops that need to be extensively analysed to ensure freedom from common mode failures and adequate allocation of requirements. Again, such loops can be illustrated by considering aircraft engines in conjunction with just four aircraft systems – Integrated Modular Avionic, Fuel Management, Electrical Power and Hydraulic Power Systems. Both hydraulic and electrical power systems are dependent on aircraft engines (and main
pumps and generators are directly powered from engines through mechanical links). On a typical aircraft, an emergency electrical power generator may be hydraulically driven – establishing a further cross dependency. A Fuel System may require electrical power for actuating fuel transfers between tanks and feeding engines. In addition to providing a rather obvious critical service to the engines, the Fuel System may contribute to cooling-down of the hydraulic fluid (thus establishing a dependency between fuel and hydraulic systems). Finally, on a modern aircraft (e.g. Airbus A380), fuel system [software] controllers may be hosted on the IMA, which, in turn requires the provision of electrical power. These dependencies are illustrated in Figure 11.

The presence of such complex dependencies requires additional multi-system analysis and has in-part motivated the new aircraft-level assessment activities (PASA and ASA) introduced in the recent revision of the ARP standards. From the perspective of WP5 of OPENCOSS, such platform-level assessment means that fully compositional certification of aircraft may not be possible as some of the safety-case claims can neither be directly attributed to nor decomposed to sub-claims that can be attributed to individual aircraft systems. These safety case claims would have to be made on the basis of integrated platform-level evidence (e.g., aircraft-level models). Furthermore, a safety case will need to demonstrate that the evidence (e.g., models) used for platform-level analyses is consistent with the evidence used for substantiating system-level claims (e.g. within the system safety case modules).

5.3 Rail

In the railway domain, the concept of compositional certification is provided for in the CENELEC standards through the following mechanisms:

- A system model with the breakdown of a system (a Specific Application) into Generic Applications and Generic Products,
- The definition of required content for the Safety Case by EN50129, which facilitates reuse by providing a standardized interface among actors (developers, assessors),
- The process of acceptance by one authority of certification delivered by another authority: cross-acceptance, which is related to reuse of certification artefacts (e.g. generic product safety cases) across different projects.
Usually, the “authority” involved is the Independent Safety Assessor (ISA) for the Generic Application, performing Cross acceptance of:

- An assessment of conformity to a standard (Certificate or ISA Report) delivered by another ISA (the most frequent case),
- A certificate granting revenue service approval (homologation) delivered by another national railway authority.

Typically, cross-acceptance is applied to Generic Products (GP) in the context of the development of a Generic Application (GA). GA’s are rarely if ever Cross accepted, and Specific Applications are never Cross accepted, being tied to a specific railway system in the hands of an operator (line, station, train).

In the following descriptions we will assume for simplicity the case of a GP being Cross accepted into a GA assessment, even if other possibilities exist.

Cross Acceptance is defined as a bottom-up process, applicable to existing products: the Cross Acceptance process does not integrate a development cycle. However, in practice, two situations are encountered:

1. The Generic Product meets the requirements of the application without any modification. The existing certification for the product can be used as-is.
2. Rework is required of the Generic Product to satisfy application requirements, a development cycle is ordered from the supplier (perhaps within the same organization) and new certification is produced before Cross Acceptance.

Figure 12 shows a typical situation involving cross acceptance, as broken down into the following five steps:

1. In the initial concept phase of the project to develop a Generic Application, a basic architecture is established. A choice of Generic Products is made based on technical and business criteria.
2. If the Generic Products are not satisfactory as they are, they may undergo a development or re-development.
3. The GPs are assessed by their traditional Independent Safety Assessor.
4. The ISA in charge of the assessment of the Generic Application performs Cross Acceptance of the GP certificates (and related reports) and produces Cross Acceptance reports.
5. These Reports are integrated into the overall GA assessment.
By definition, the GP lifecycle is distinct from that of the GA. There is no integrated flow through a V-cycle of requirements allocation from GA to GP. To compensate, ad-hoc verification must be performed to show that the requirements of the GA are indeed satisfied by the GP that has been taken off the shelf.

The concept of a Safety Case contract provided by a Generic Product can be stated informally as follows:

This Generic Product will fulfil its specified functions with adequate safety assuming operation within specified environmental constraints and assuming satisfaction of exported safety requirements.

These essential features of the Generic Product must be documented as required by EN50129 in the Safety Case. This leads directly to the requirements for cross acceptance of a certificate, which can be performed if the following conditions are fulfilled:
• The GP is fit-for-purpose: the GP satisfies the relevant requirements of the GA.
• There is similarity in the normative reference of the Generic Product and the target Generic Application.
• The environmental conditions for operation of the GP match the environment of the target GA.
• The exported Safety Related Application Conditions (SRACs) from the GP can be or have been satisfied by the GA.

The conclusions of the certificate are positive and delivered by a trusted authority.

It is important to note that whilst the overall “architecture” of the compositional certification process in the railway domain is well-understood, the details of the process vary considerably between jurisdictions of different national authorities and, at times, the jurisdictions of individual ISAs. Most importantly, from WP5 perspective, there is considerable variation in the practices employed with respect to documenting various certification artefacts and data (such as SRACs, environmental conditions, key properties offered by GPs as well as GP and GA safety cases as a whole). The certification process is currently conducted predominantly on the basis of textual reports; whilst EN50129 attempts to instil some common structure, in practice there is considerable variation in the format and even content of the reports produced by different organisations. The purely textual (and, often, free-text) format not only makes document management and retrieval overly labour intensive, it also means that consistency between different safety case parts and validity of composition (see the last set of bullet points above) have to be established through an entirely manual and informal inspection process.

5.4 Other domains: Defence

The Industrial Avionics Working Group (IAWG) was setup by the Ministry of Defence (MoD) in the United Kingdom, to look at ways to develop modular and incremental certification. Modular (or, in the present document’s terminology, compositional) certification refers to the ability to build a system out of well-defined modules, whereas incremental certification refers to the ability to change, update or add a module to an existing system. The MoD is particularly keen to reduce the time it takes to update military equipment and to reduce certification costs. One of the most frequent comments about recertification costs is that they are often proportionate to the size of the system as a whole rather than the size of the change (Fenn, Hawkins, Williams, Kelly, Banners, & Oakshott, 2007). This means that the cost of re-certification of a system when there has only been a minor change can be extremely high. Also, the alternative of simply not being able to re-certify means that legacy parts need to be kept and maintained at a major expense. The lifespan of some military equipment (e.g. aircraft in service for tens of years) makes this very difficult and has led to methods such as deep freeze of microprocessors that are no longer manufactured.

In (Fenn, Hawkins, Williams, Kelly, Banners, & Oakshott, 2007) members of the working group discuss some of the results and findings of the IAWG when looking at military IMA (see section 6.1.1). The paper outlines their approach with four major steps, which are taken from the point of view of a well-defined system:
1. Identify change scenarios for the system - this helps identify which components may change, the likelihood of a change, and hence the best-suited design of the system to minimise the impact of the change.

2. Define the safety case architecture (the authors use GSN) - this is based on the system architecture, and also contains the likelihood of changes. Basic principles of high-cohesion, low-coupling (between argument modules), well-defined interfaces and information hiding are advocated.

3. Dependency-Guarantee-Relations (DGRs) - these are the equivalent of software contracts, i.e., pre and post conditions. Their definition helps guarantee inter-operability. So far only software has been examined; hardware and other contracts need further research.

4. Finally, generate the safety argument with modules showing how DGRs are met, and a higher-level argument that shows how the DGRs meet and support hazard mitigation and prevention.

It is worth noting that the group notes the difficulty in modularising evidence as an ongoing challenge, and also notes several appropriate and inappropriate situations where the technique should perhaps not be used.

The findings of the IAWG are very useful for the OPENCOSS project. Even though they were for a relatively fixed type of system design, they considered in depth the impacts of changes, and discovered that many limitations and trade-offs were required. OPENCOSS has a wider ambition, and should consider how these results are relevant in more complex, cross-domain reuse contexts with less strict design.
6 State-of-the-art

The literature and the survey results presented in section 2 show that compositional certification is organised into a number of “streams” of approaches. Section 6.1 describes design modularity to standardise architectures, interfaces and partitioning for safety critical systems. Section 6.2 describes certification contracts, in order to define the behavior or properties of a component (or subsystem) under a given context. Safety cases, to enable separation of certification data (as an alternative to a monolithic approach), are described in section 6.3. Section 6.4 mentions other research work relevant for WP5 and, finally, in section 6.5, some tools for compositional certification are introduced.

6.1 Design modularity: Standardised architectures, interfaces and partitioning

One important way to support compositional certification is via the use of standardised well-defined software architectures. For example, the architecture can have well-defined software interfaces, so that different developers and manufacturers can create their own versions of products that support them but may offer different functionality (e.g. applications using an OS). In addition, the architecture may offer robust ways to prevent software from sharing (and potentially corrupting) resources except in a rigid and managed way. The following section describes two well-known examples, IMA and the AUTomotive Open System ARchitecture (AUTOSAR).

6.1.1 Integrated Modular Avionics

IMA, (or, occasionally, Integrated Modular Systems - IMS) is the term used for a heterogeneous network of computer nodes aboard an aircraft. Previously, aircraft systems were federated, with their own dedicated computing resources, but this led to extremely complex and varied spares being required, and made it very difficult to upgrade systems and software. IMA uses a standardised three-layer stack architecture as shown in Figure 13. There are two main types of IMA - military IMA as described in the ASAAC standard (Defence M. o., 2008) and civil IMA as described in the ARINC 653 standard (ARINC). Here we describe an overview of shared features that support compositional certification.
IMA has several features aimed at facilitating compositional and modular certification:

- **Strict partitioning of resources**
  - Processing partitioning: by strict scheduling that is guaranteed and easy to predict, but is not optimal in terms of performance. Thus timing analysis is supported for certification.
  - Memory partitioning: data cannot easily be shared between applications, again this may affect performance but makes it easy to argue/demonstrate that there is no data corruption between software of different integrity levels.
  - Network partitioning: as above, access to the network is also partitioned to assist in analysis and demonstration of key properties such as latency.

- **Hardware transparency** - the use of a hardware interface layer, abstracting hardware (processor) specific features from the application and an operating system imply that moving them to a new processing platform should require no change to the code. Obviously, some performance metrics may change, but avionics systems are no longer tied to out of date legacy hardware.

These features also assist in incremental change and re-configurability of IMA. For example, applications can be easily added to the platform as long as existing partitioning boundaries are not affected.

Some aspects that may need to be considered with respect to compositional certification of IMA (or similar systems) are:

- The definition of a clear incremental, and probably iterative, certification process (Yong, Zexin, Xupo, & Linag, 2010).

- The development of safety cases (Bate & Kelly, Architectural Considerations in the Certification of Modular Systems., 2003), which, in the IMA context, should address both present and future configurations of a system. This is related to and can be addressed by means modular safety cases (see also section 6.3 of the present document).

- Management of multiple levels of requirements, development of safety assessments, management of system configurations with multiple and reconfigurable components, identification of parts via electronic means, software assurance, assurance of complex electronic hardware, component qualification for a given environment (i.e., context), monitoring of system health, definition of roles
and responsibilities for all the stakeholders, and integration of multiple components (Lewis & Rierson, 2003).

- The role of different suppliers in developing a safety case for an IMA-based system (Wilson & Preyssler, 2009).

### 6.1.2 AUTOSAR

AUTOSAR is a (similar to IMA) software architecture paradigm, being used in the automotive industry. Various parts of the AUTOSAR specification can be downloaded from the standardisation body’s website (AUTOSAR project). The standards have been developed by a consortium of many of the major automobile manufacturers and component suppliers. On the one hand it allows for interchangeable components, meaning that suppliers can provide parts to multiple manufacturers; on the other hand it guarantees functionality and performance to the vehicle manufacturers.

As well as design architecture, AUTOSAR defines a development methodology. This is not a very strict set of processes, rather some high level guidance as to dependencies between tasks and workflows. This may be of concern for OPENCOSS, particularly if we are planning on re-using software across domains. It may be necessary to show a degree of compliance to the process somehow.

Finally, tool interoperability is also specified in AUTOSAR. Again, this is relevant to OPENCOSS as it may be necessary to ensure that the OPENCOSS platform is compliant with standards and requirements, or can at least operate with tools that are compliant.

### 6.2 Design modularity: Design-by-contract

A contract-based design in system engineering involves the following cornerstones:

- Components and subsystems
- Contracts
- Interaction points

A component is a modular unit of a system, generally not further decomposed in other components (see also ‘Compositional certification specific definitions’ above). A subsystem is a modular unit of a system, which can be decomposed in other subsystems and components.

In response to the above challenges, new contract-based system design paradigms have recently emerged. Under many such approaches, a contract consists of a number of “<Assumption, Guarantee>” pairs. In each such pair, a guarantee is the specification of a behavior or property that the component is declared to meet provided components context satisfies the declared assumption.

A contract is usually associated with a component via interaction points, or interfaces. Figure 14 shows two components – ComponentA and ComponentB – that are associated with two contracts - Contract1 and Contract2 (respectively). Broadly speaking, an interaction between ComponentA and ComponentB is feasible only if the two contracts match.
A contract-based approach allows abstracting from the details of component implementation (such as internal structure, modeling and – for software – implementation language) whilst capturing key properties that are important for correct interaction between components in the wider system context.

Another important principle of modern design-by-contract approaches is “separation of concerns”. This refers to the ability to specify, analyse and/or verify a system from a perspective of a number of viewpoints in relative isolation from each other. Separation of concerns facilitates reuse of design artifacts and relies on the notions of modularity and encapsulation. In simplified terms, separation of concerns allows to divide \(<\text{assumption},\text{guarantee}>\) pairs within each component contract into a number of groups (often called “views” or “viewpoints”) linked to particular aspects (typically, quality attributes) of the system (such as performance, safety, security, etc.). Pairs that fall into the same group are often referred to as “component contract ...” further qualified by the corresponding view (e.g. “security contract of the component”).

Among the main advantages of a contract-based approach, we point out two properties, known in the literature as \(\text{composability}\) and \(\text{compositionality}\) (Sifakis, 2005), (Sifakis, 2005).

- Composability is the ability to preserve a component’s properties “when it is integrated in an environment”
- Compositionality is the ability to infer system properties from component properties, that is, “composability guarantees that if the components of a system meet a given property then this property is preserved by composition”

If we take together \(\text{composability}\) and \(\text{compositionality}\) properties, we achieve what is technically known as a \(\text{correctness-by-construction}\) approach to software engineering (Sifakis, 2005).

The following sub-sections will briefly discuss some examples of this contract-based approach to system engineering.

**Interface Automata.**

The article “Interface Automata” (Alfaro & Henzinger, 2001) defines a solid mathematical structure for software component interfaces (Alfaro & Henzinger, Interface automata in Proc. 9th Annu. Symp. Foundations of Softw. Eng, 2001). The authors highlight the leading role of temporal attributes in a software development and deployed them in interfaces. In this framework, the authors define an
automata-based language that exploits the notion of contracts, where, an assumption is an ordered sequence “in which the methods of a component are called”, and a guarantee is the order “in which the component calls external methods”. A key notion is the interface composition that leads to the smaller automata, which should meet constraints related to the environment and the behavior.

Assert EU Project.
Under approach developed by this project (ASSERT project), assumptions of the component contracts are modeled as provided interface and guarantees – as required interface (Cancila, Passerone, Vardanega, & Panunzio, 2010) as illustrated in Figure 15.

An interaction between components can occur only if their respective contracts match and if the transitive closure over assumption is met. To illustrate this consider a three-component example in Figure 16.

Let’s first consider only Component A and Component B of Figure 16:
- Component A guarantees some services (PI_1) with attributes declared in the contract (e.g. deadline of a task, safety integrity level and mean-time between failures) if its assumption, RI_1, is satisfied.
- This required interface RI_1 is satisfied by the guarantee PI_2 provided by Component B, (assuming the two contracts locally match).

The match is not in itself sufficient to conclude that the overall system guarantees PI_1. This is because PI_2 requires RI_2 to be satisfied. However, if we now consider Component C we can see that its interface PI_3 satisfies RI_2. Overall, we now have a chain from PI_1 to PI_3 where the last element (PI_3) is not dependent on any further required interfaces. Consequently, the system guarantees PI_1. A theory, which exploits hyper graphs, supports the transitive clotrure and other safety-related properties, such as liveness (Cancila, Passerone, Vardanega, & Panunzio, 2010).

CHESS Project
The CHESS project has developed the issues introduced in the ASSERT project further (Vardanega, 2009). The CHESS project has adopted the same contract-based approach and the same underlying mathematical
structure of ASSERT. While the Assert project was focused on spatial application only, the CHESS project has expanded the application of the approach to aero-spatial, telecommunication and railway application domains. thus strengthening results on both contract-based approach and mathematical structure.

SPEEDS EU project
In the SPEEDS project (SPEEDS project), a component is called rich component. It is described by a set of ports and variables. The behavior of a component can be described, e.g., by automata. A contract is a pair “promise, assumption” and characterizes a component. “An assertion is a context under which the component might be used” (Passerone & all., 2009) and a promise is a possible behavior of the component under the context, which is declared in the assumption. Finally, a component satisfies its contract if the promise is met under the context.

SPEEDS mathematically defines four main pillars: refinement of a context; substitutability or dominance; conjunction of contracts, and parallel composition. Among them, the key concept is substitutability or dominance, because it allows us a hierarchical structure of a system. Broadly speaking, a contract dominates another contract if it provides more contexts and less promises. Formally, Contract C_0 dominates Contract C_1 if the following conditions are hold together:

- the assumption of C_0 includes them of C_1
- the promise of C_0 is included in that of C_1

Then, Contract C_0 can be substituted with Contract C_1.

More information on mathematical structure can be found in (Passerone & all., 2009) and (Benveniste, Caillaud, Ferrari, Mangeruca, Passerone, & Sofronis, 2008).

6.3 Compositional safety cases
Safety cases either presented textually or by means of graphical notations – such as GSN (Goal Structuring Notation Working Group, 2011) or CAE (Adelard) – can be constructed as monolithic artefacts. However, being quite complex and lengthy documents the monolithic structure often does not reflect the key characteristics of a system and safety case development processes. Furthermore, monolithic safety cases pose a number of problems including those of managing safety case change and evolution (e.g. as a result of changes to product itself or, its operating or regulatory environments) and managing traceability between ‘parts’ of the safety case on the one hand and product ‘components’, stages in the development / safety lifecycle process, and scopes of responsibilities of various supply chain / engineering stakeholders on the other.

This section explores the notion of modular and compositional safety cases, an alternative to the monolith approach. The first sub-section briefly reviews the key concepts of modular safety cases.

---

5 Readers interested in the details of the two notations are referred to the cited sources or, alternatively, to OPENCOSS deliverable D4.1 (“Baseline for the Common Certification Language”). From now on we will assume basic knowledge of GSN (which most readers will find self-explanatory)
The second subsection discusses different forms of safety case modularity and links those to the notion of compositional certification (i.e. the key objective of WP5). The section concludes with a discussion of some of the R&D challenges in the area of modular safety cases & compositional certification that may need to be considered by OPENCOSS project as a whole and WP5 in particular.

We use the Goal Structuring Notation throughout this section to illustrate the concepts presented. However the principles discussed are largely common to all safety cases regardless of the way to present them. The only restriction that we pose is that a safety case can be defined as:

“A structured argument, supported by a body of evidence, that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given environment” (Defence U. M., 2007).

### 6.3.1 Key principles of modular safety cases

The notion of modular safety cases has been introduced to facilitate effective development and maintenance of a safety case. Inspired by the notions of modularity in software and system engineering disciplines, modular safety cases are divided into coherent units (modules) with well-defined interfaces. The notion of modular safety cases is based on three key concepts:

- Safety-Case Modules
- Safety-Case Architectures
- SafetyCase Contracts

#### 6.3.1.1 Safety Case Modules

Safety Case Modules are parts of an overall safety case containing part of an argument and relevant citations of evidence. The well-defined safety case module typically addresses a logically cohesive set of objectives and minimises the amount of cross-referencing to, and dependency on, other modules (Kelly, “Using Software Architecture Techniques to Support the Modular Certification of Safety-Critical Systems, 2006). A safety case module may correspond, among other things, to an interrelated set of safety engineering activities (e.g. stage of a safety lifecycle), scope of responsibilities (contractual or otherwise) of a particular engineering organisation (e.g. a supply chain stakeholder), well-defined sub-system or equipment used within the overall safety-critical platform.

The key principle behind safety case modules is that of information hiding: unless alternative provisions are made (as discussed below), claims internal to the module and the relationships between such claims are not visible from other modules. However, safety case modules clearly need to rely on or take account of some of the information contained in other modules. Therefore some of the elements of a safety case module must be made visible on module’s *interface*. Key elements of the publically-visible interface of a safety case module include (Kelly, “Using Software Architecture Techniques to Support the Modular Certification of Safety-Critical Systems, 2006):

- Objectives (argument claims) addressed by the module
- Evidence presented within the module
- Context defined within the module
- Reliance on objectives (claims) that should be addressed elsewhere
• Reliance on evidence that should be presented elsewhere
• Reliance on the context defined elsewhere

To some extent, this is analogous to the design-by-contract principles discussed in section 6.2. In particular, the first three items of the above list are similar to the guarantee of a component, whereas the last three items are analogous to the assumptions that are made within (i.e., relied upon) the component (and that should be fulfilled elsewhere). Similarly, a safety case module can only substantiate its own public objectives provided that the relied upon objectives (as well as contexts and evidence) are appropriately substantiated by other modules. Figure 17, illustrates the notion of the safety case module and module interface.

![Safety Case Module interface](image)

Figure 17: Safety Case Module interface

Finally, Figure 18 shows an example of a simple (uninstantiated) safety case module in GSN. In this diagram, the Goal SysAccSafe is marked with a small module symbol to indicate that this is a publically visible goal of the module. The module relies on two away goals (i.e., goals declared in other modules): FunctionsInd (declared in IndependenceArg) and FnBSafe (declared in FnBArgument). Finally, Goal FnASafe requires support from another module (FnAArgument); however, it could not be yet linked to an individual away goal (e.g., because the module FnAArgument has not yet been fully developed).
6.3.1.2 Safety Case Architectures

As the structure for safety cases of safety critical systems is typically complex, it is often necessary to obtain high level view of the overall organisation of the underlying argument. Whilst modules provide a mechanism for encapsulating parts of the overall safety case, the relationships between those modules also need to be described (often, for ease of communication, in a graphical form).

The problem is, again, analogous to some of the issues encountered in the field of software engineering. There, to facilitate the management of complexity in large-scale software intensive systems, the concept of software architecture has emerged. Bass et al define software architecture as “The structure or structures of the system, which comprise software components, the externally visible properties of those components and the relationships among them” (Bass, Clements, & Kazman, 2003)

Similarly, safety case architecture can be defined as “the high level organisation of the safety case into components of arguments and evidence, the externally visible properties of those components, and the interdependencies that exist between them” (Kelly, Managing Complex Safety Case, 2003). The notion of safety case modules, discussed previously, allows representing safety case architecture through a high level view of the interconnections between safety case modules, see Figure 19.
Similarly to software architectures, it is also possible to specify general criteria for effective structuring of safety case architectures and to formulate the ‘good practice’ in safety case decomposition into modules (Bates, 2003) (Fenn, Hawkins, Williams, Kelly, Banners, & Oakshott, 2007):

- Modules must exhibit **high cohesion**, meaning that objectives of the safety case module must be well focused, logically related and naturally consistent.
- There must be **low coupling** between the modules, thus cross referencing between different modules should be minimised as much as possible.
- Safety case **modules must have well-defined interfaces** and all modular dependencies must be captured.

To maximise benefits from modular structure, the principle of *information hiding* should be followed. In other words, only the minimum necessary information should be made public on the module’s interface and all other information should be hidden from (i.e., not made available for referencing from within) other safety case modules.

It is important to stress that the concepts similar to that of safety case architecture can be found in a number of safety standards. For instance the safety case (and individual “reports”) structure prescribed by CENELEC 50129 standard is essentially a safety case architecture with individual parts (such as Quality Management Report, Safety Management Report and Technical Safety Report) corresponding to a large extent to a concept of safety case module\(^6\). Furthermore, explicit mention of “related safety cases” (i.e. Part 5 of the safety case structure prescribed by EN50129) in the standard recognises the need for capturing the relationships between safety cases (and, in our terminology, safety case modules).

When safety case architecture is not explicitly prescribed by a standard, it can often be ‘reverse-engineered’ based on the advocated process (e.g. the safety lifecycle), requirements and guidance. For instance, Figure 20 shows a safety case architecture [pattern] developed in the MISSA project (see section 10.1 on the basis of the ARP4754a document \(^7\).

---

6 The correspondence is even stronger for sections of the Technical Safety Report listed in Section 5.4 of EN50129.
7 The architecture pattern was developed by Linling Sun, PhD candidate with Department of Computer Science, the University of York.
As was mentioned before, modules of the safety case are not completely independent from each other and, whilst dependencies between the modules must be kept to the reasonably practicable minimum, a modular safety case framework must provide some means for capturing cross-references. This can be achieved through the notion of the safety case contract. To some extent analogous to the notion of
software contracts, safety case contracts establish traceability links between the goals (claims), contexts and evidence relied upon by a module and those publically offered (addressed, defined and presented respectively) by other modules.

Within GSN, the simplest representation of the contractual relationship between two safety case modules can be achieved by using away goals as previously illustrated in Figure 18. This allows one module to refer to an element (goal or context) defined on the interface of another module (Goal Structuring Notation Working Group, 2011).

However, this can be seen as ‘hardwiring’ safety case modules and, to some extent, violating the information hiding principles, since the internal module structure would have to be ‘aware’ of elements – albeit public – in another module. More importantly safety case modules may need to be developed concurrently and the goals of one module may not be available at the time when reliance of another module needs to be specified.

The IAWG reviewed the support offered by GSN to construction of modular safety cases and specification of safety case contracts, see also section 5.4. Having found many limitations of the initial solution based on a tabular format for contracts, they proposed (Fenn & al, Safety Case Composition Using Contracts - Refinements based on Feedback from an Industrial Case Study, 2007) a generalisation of the contract concept and an approach for contract specification within the GSN notation itself. One of the advantages of this approach is the ability to specify more complex relationships between goals declared in different modules than simple one-to-one “solved-by” and “in-the-context-of” mappings. The solution essentially allows contracts to contain additional arguments that do not ‘belong’ to any module and that, instead, provide ‘glue’ between the modules. For example, Fenn et al have also proposed a standard GSN pattern (Figure 21) that can be used for some of the contracts.
Finally, whilst there are similarities between safety case and design (including software) contracts, it is important to recognise that the two are fundamentally different. Software contracts record dependencies between components whereas safety case contracts resolve dependencies between assurance arguments. In the context of compositional certification (see next section), design contracts and safety case modules are often interdependent. Design contracts identify key properties of the components that must be fulfilled in order for the software system as a whole to provide adequate (including safe) service, whereas safety case modules contain arguments and evidence that demonstrate that those key properties have been achieved.

6.3.2 From modular safety cases to compositional certification

On its own, modularity simply provides a means for structuring of the safety case. It does not necessarily affect the safety case development process – modular structure can be post-imposed onto a safety case that has been developed as a single artefact. This may be beneficial for the review of the safety case (since the modular structure will provide the basis for divide-and-conquer approach) as well as for the maintenance of the safety case (where a modular structure may be devised on the basis of anticipated change scenarios in order to minimise the number of modules affected by the most likely changes).

However, safety case modularity facilitates compositional safety case development. Under this approach, the overall safety case integrator (typically – the overall safety critical platform integrator) can first develop a safety case architecture by identifying key parts of the safety case (the safety case modules) and their interrelationships. Safety case modules can then be developed relatively independently and, for instance, by different engineering teams and organisations. The integrator composes the safety case from the
modules as they become available by specifying safety case contracts. It is important to stress that contract specification is a non-trivial task that requires review of the safety case modules as well as checking consistency between goals and contexts declared in different modules.

As was previously discussed, the ‘hallmarks’ of a good safety case architecture are close alignment between modules on the one hand and the structure of the engineering and assurance processes on the other, as well as low coupling between - and high cohesion within - the modules. The modular structure of the safety case, however, does not necessarily have to align with the structure of the system. Other strategies for modularising safety cases may exist and be equally or more beneficial in different circumstances. For instance, the architecture previously presented above in Figure 20: Sun’s Reference Safety Case Architecture for ARP4754a has been based on the safety lifecycle of the ARP4754a standard rather than the decomposition of the aircraft into its systems, subsystems and key equipment.

Nevertheless, compositional safety cases are most frequently motivated by the notion of compositional certification whereby safety case modules correspond to the key equipment or components of the system and where the responsibility for their development (and, sometimes, obtaining the regulatory approval) is assigned to the corresponding supplier.

It is important to note that it is unlikely that an adequate safety case can be constructed exclusively by composition of component-related safety case modules. Firstly, the integration of individual modules e.g., through safety case contracts may require significant non-trivial argumentation and justification. Secondly, a system safety case is likely to address a number of transversal issues that cannot be attributed to individual components.

### 6.4 Other research work relevant for WP5

At this stage of OPENCOSS, the R&D work concerning modular safety cases and safety case modules, architectures, and contracts is regarded as the most relevant to WP5. Nonetheless, it is important to analyse more publications in order to obtain a wider perspective on the possible solutions and challenges for compositional certification. This section briefly reviews some peer-reviewed publications in order to complement and widen the review of the state of the art presented so far. The overall objective of this section is to provide more insights and details that might need further consideration and investigation during the development of WP5.

Many researchers have studied the use of commercial off-the-shelf (COTS) components in critical systems and its implication for safety certification with the aim of showing the adequacy and feasibility of this approach (e.g., (Kesseler, 2008)). A clear challenge is the lack of information about the development process, thus provision of product information as certification evidence (see also OPENCOSS D6.1) has so far been the focus of the research (Redmill, 2004).

Safety case contract-based approaches have been proposed for certification of COTS-based systems (Ye & Kelly, 2004), as well as explicit methods for addressing certification of this type of systems (Ye & Kelly, 2004). For goal-based standards, a framework for dealing with and providing safety evidence was proposed in (Menon, McDermid, & Hubbard, 2009), which addresses failure consequences and integration...
assumption, adequacy, degree of protection, inputs and data flows, and resource availability and interference. Although almost 10 years old, indications regarding the use of COTS in different safety standards and domains has been discussed in (Kesselrer E., 2003). At that moment, the author concluded that the standards should be adapted and thus updated in order to meet industry needs regarding the use of COTS.

Related to the work discussed in the two previous paragraphs, some authors have studied the implications and challenges of software reuse and composition. The main conclusions that can be extracted from these works are:

- Some challenges that can be faced with this approach are the management of component interfaces, component abstraction, the activities performed by integrators, traceability, and certified vs. certifiable components (Akelholm & Land, 2009).
- When dealing with open source software, and given that trust in the parties involved in its development is difficult to argue for safety assurance and certification, provision of product information such as testing and formal verification results seems to be the only possibility (Kakarontzas, Katsaros, & Stamelos, 2010).

Other authors that explicitly dealt with issues related to WP5 have focussed on the following issues:

- Modular certification based on rely/guarantee contracts (Blow, Cox, & Liddell, 2005), suggesting the convenience of a product-based approach and thus of product information as evidence.
- The use of generic safety cases for modular certification, advocating the reuse of their arguments (Althammer, Schoitsch, Sonneck, & Vinter, 2008) and focusing on architectural aspects (Althammer, Schoitsch, Eriksson, & Vinter, 2009). The latter work is in line with (Wu & Kelly, 2007).
- Provision of a tool-supported semi-automatic integration method for modular safety cases (Zimmer, Bürklen, Knoop, Höfflinger, & Trapp, 2011)
- Reuse of safety cases (Bush & Finkelstein, 2001), indicating needs such as classification of the information reused and the provision of the information model used/followed in a safety case.
- The definition of a process for modular safety assessment and analysis (Bate & Conmy, 2005), aiming to, among other purposes, generate safety assurance contracts.

Some of the publications cited above will be examined in detail in task T5.1 of OPENCOSS project to identify any past work that can usefully feed into WP5 of the project. In addition to considering the substantial proposals made by the authors of those papers, the work will be reviewed to identify any follow-up work and to ascertain the level of maturity reached by the research since original publications.

### 6.5 Compositional Certification tooling

The questionnaire results as described in Section 2 did not result into tooling that partners have been using in practice for compositional certification. This may be attributable to the fact that the notion of compositional certification is relatively novel and that industrial practices in this regard vary greatly both within and across the boundaries of industrial domain. Nevertheless, Table 2 introduces some tools that, in WP5 partners view, may support compositional certification.
<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCE (Panesar-Walawege, Sabetzadeh, &amp; Briand, 2011)</td>
<td>Use of UML profiles or metamodels, in conjunction with OCL-constraints, can be a candidate solution/technology for composability assessment</td>
</tr>
<tr>
<td>EvidenceAgreement (Falessi, Briand, Sabetzadeh, Turella, Coq, &amp; Panesar-Walawege, 2011)</td>
<td>It might serve as a basis for agreement with certification authorities upon composability</td>
</tr>
<tr>
<td>Programatica (Hallgren, Hook, &amp; Jones, 2004)</td>
<td>Authors claim it supports “compositional certification”, although in a very primitive, simplistic way</td>
</tr>
<tr>
<td>Modus (Sabetzadeh, et al., 2011)</td>
<td>It provides an environment for construction of goal-based arguments, to elicit probabilities from experts about goal satisfaction, and to propagate the elicited probabilities for reasoning about the satisfaction of high-level safety goals.</td>
</tr>
<tr>
<td>SafeSlice (Nejati, M., Falessi, Briand, &amp; Coq, 2012)</td>
<td>It provides a SysML-based environment for establishing traceability links from safety requirements to design models and for extracting fragments of design that are relevant to safety requirements. It establishes a link between traceability and composition.</td>
</tr>
</tbody>
</table>

Table 2: Suggestions for tooling including description for compositional certification
7 Overview of projects and initiatives related to compositional modular certification

As part of the work on establishing the baseline for WP5 (reported in this document), authors have reviewed a number of initiatives related to the subject matter of the work package (compositional certification, modular certification, contract-based certification and component-based certification) and reported by OPENCOSS partners. These initiatives are summarised in Table 3 and presented in Appendix 10 (where, for each initiative, its purpose, the common characteristics with OPENCOSS and WP5, and the contribution of WP5 beyond the considered initiative are summarised).

<table>
<thead>
<tr>
<th>Initiative</th>
<th>Project/initiative name</th>
<th>Possible useful input for OPENCOSS WP5</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP research</td>
<td>ASSERT</td>
<td>Tested contract-based approach to design to certification issues</td>
</tr>
<tr>
<td></td>
<td>MAENAD</td>
<td>The concept of an evolving architecture</td>
</tr>
<tr>
<td></td>
<td>MISSA</td>
<td>The composition of modules related to safety assessment performed at different stages of the safety lifecycle promoted by the de-facto safety standards in the civil aviation sector</td>
</tr>
<tr>
<td></td>
<td>MODSafe</td>
<td>Identification of current safety approaches, comparison of life cycle approaches, a proposal for safety life cycle approach and different approaches for acceptance, approval and certification.</td>
</tr>
<tr>
<td></td>
<td>SPEEDS</td>
<td>Reuse of mathematical and model based-design approach viewpoints</td>
</tr>
<tr>
<td>ARTEMIS Joint Undertaking</td>
<td>CESAR</td>
<td>Reference to composability criteria in general</td>
</tr>
<tr>
<td></td>
<td>CHESS</td>
<td>Reuse of the identification and precise specification of properties (in within models) related to safety.</td>
</tr>
<tr>
<td></td>
<td>MBAT</td>
<td>Reuse of models and artefacts for development, integration and delivery of the high-quality products in the transportation domain</td>
</tr>
<tr>
<td></td>
<td>RECOMP</td>
<td>To provide insights into the customization and upgradability of safety-critical parts, for changes in modules within compositional systems.</td>
</tr>
<tr>
<td></td>
<td>SafeCer</td>
<td>Potential support for system safety arguments based on arguments and properties of system components and to provide support for the generation of corresponding evidence in a similar compositional way.</td>
</tr>
<tr>
<td>ITEA R&amp;D</td>
<td>EVOLVE</td>
<td>Although EVOLVE focused on V&amp;V, it dealt with composition, through product development iterations.</td>
</tr>
<tr>
<td>Other</td>
<td>Flanders’ DRIVE ASIL</td>
<td>To make use of unification of the development processes, and to allow certification according to different standards.</td>
</tr>
</tbody>
</table>

Table 3: Initiatives including possible useful input for compositional modular certification

Overall, it should be noted that whilst a number of the initiatives have considered issues relevant to the WP5, the focus of the vast majority of them is on compositional design or a broader issue of compositional development. Only two initiatives were found to have the scope that is similar to that of WP5 – SafeCer
projects of the ARTEMIS Joint Undertaking (see Section 10.2) and UK Ministry of Defence funded IAWG (see Section 5.4 along with some of the discussion in Section 6.3.1.3). These two projects will be studied in more detail as part of the work on eliciting detailed requirements for WP5. Furthermore, to ensure completeness, WP5 partners will perform a further iteration of the search for related projects.
8 Conclusions

This document has defined the baseline for the work to be undertaken by OPENCOSS WP5, with respect to compositional certification objectives. The specific objectives of WP5 are to:

1. Understand how to capture each component’s assurance contract and how to propagate the contracts for certification acceptance by other components
2. Identify the necessary contractual information
3. Consider emergent properties or unexpected interactions which may arise during safety case integration

The baseline has been defined through considering:

- Business drivers for compositional certification
- Relevant requirements contained within the safety standards
- Existing industrial practice (as reported, formally and informally, by the OPENCOSS partners)
- State-of-the-art (including ongoing research initiatives) in the discipline of safety assurance

The key findings (with respect to the four groups of issues above) are summarised in the next subsection. We conclude this document with a summary of key areas for future work (section 8.2) and description of the next steps that need to be performed by task T5.1 of the project (section 8.3).

8.1 Summary of findings

Through survey, inspection of OPENCOSS D2.1 report and various discussions with the project consortium partners as well as within our wider networks of collaborators, the authors have established that there are significant business drivers for the compositional certification of safety-critical systems in aviation/avionics, automotive and railway domains. Whilst different industrial organisations may express those drivers and motivations differently, they tend to centre on a fundamentally similar set of issues:

- Maintaining the certification costs and efforts in a light of continuous changes to the business and development processes such as increasing degree of component re-use, subcontracting of development activities and the increasing complexity of the supply chain.
- Minimising, wherever possible, the costs of certification (whilst at least maintaining the rigour of the overall process and safety of the systems) through enabling systems integrators to take ‘certification credit’ for previously assessed / certified components.
- Reducing the re-certification effort and costs for components reused across the boundaries of industrial domains and jurisdictions of different standards, regulators or assessors (when components have been previously certified in another domain/jurisdiction).
- Controlling the cost of re-certification following system change and, in particular, ensuring that re-certification efforts are, at worst, proportionate to the extent of change rather than to the size of the whole system.
Given the above motivation, we have examined the relevant guidance and requirements contained within safety standards to ensure that the project takes those into account. Overall, whilst (unsurprisingly) no standards require certification to be performed in a compositional manner, virtually all acknowledge the industrial need for compositional or modular certification of complex safety-critical systems. There is, however, significant variation between the standards:

- Whilst in some sectors (e.g. automotive and railway) standards explicitly define concepts related to compositional certification (e.g. safety element out of context and generic safety cases respectively), in others (e.g. aviation) – the recognition of non-monolithic certification process is much more implicit.
- Guidance contained within standards of different industrial sectors adopts different conceptual perspectives on the nature of modularity with respect to certification. Those perspectives can often be seen as complementary.

Across industrial domains the guidance on performing compositional certification contained within standards requires significant interpretation, and never constitutes a ‘complete manual’ that can be followed by the system integrators and component developers. Consequently, it is crucial to consider what practices are currently adopted by the industry to enable compositional certification within the context of standards’ requirements and guidance.

In discussions with project partners, we have established that industrial practices with respect to compositional certification vary to great extent not only between different industrial domains, but between different organisations within each domain. However, across all domains, and with only rare exceptions (such as IAWG consortium in the UK, see Section 5.4), certification artefacts are captured by means of mainly textual reports with reconciliation of artefacts typically performed in fully manual manner and adequacy of composition assured through informal manual inspections. This often results in lengthy and labour intensive process. The time and cost overheads associated with the compositional certification are further exacerbated by the lack of support tools. Currently, companies tend to adapt general document management tools to support management of certification artefacts; these tools are often bespoke and inconsistent across the supply chain.

Finally, in establishing the baseline for WP5 we have considered a number of research initiatives and academic publications to clarify the state-of-the-art in compositional certification. Our key findings in this regard are:

- Whilst the overall discipline of compositional and contract-based development is reasonably mature – with a large number of past and ongoing initiatives – most of the R&D work today has focussed on design-by-contract notion and standardisation of system architectures, rather than on the topic of compositional certification.
- Standardised well-defined software and system architectures (such as IMA or AUTOSAR) can support compositional certification.
- Whilst many general principles of design-by-contract paradigm are relevant to compositional certification, they require substantial adaptation and the two disciplines are distinct.
- The most mature research in the field of compositional certification is based on the notion of compositional safety cases; this relies on the concepts of “safety case module”, “module’s public interface” and “safety case contracts”.

Finally, in establishing the baseline for WP5 we have considered a number of research initiatives and academic publications to clarify the state-of-the-art in compositional certification. Our key findings in this regard are:
- Safety case architecture forms a basis for compositional certification.
- Within the framework of safety case architecture, modules can only be properly composed if assumptions (and other contextual information) associated with the arguments of and with the evidence cited by different modules are compatible.

8.2 Areas requiring further R&D work

Although a number of past and ongoing research projects and even a number of industrially-led initiatives have worked on compositional safety case development, a number of challenges in this area still need to be addressed. In this section we list some of the key challenges as perceived at this stage by OPENCOSS WP5 partners.

8.2.1 Matching explicit context and assumptions

As stated above, when composing a system safety case from individual safety case modules, it is necessary to match claims relied upon by a module to claims ‘exported’ (i.e., supported and publically declared) by other module(s). Also, contexts of the composed modules (expressed in GSN by context, assumption and justification declarations) need to be examined for consistency.

The matching and consistency checks are non-trivial due to the complexity and diversity of the declarations (formulated using natural language), as well as due to the sheer number of declarations. Furthermore, these tasks are currently left entirely to safety engineer’s judgement. As the goals/claims, context, assumption and justification are declared purely in free text, this makes them less accessible to software tools (more information in section 6.3.1.2).

Whilst full automation and formalisation is unlikely to be feasible, OPENCOSS WP5 should attempt to provide some support to the tasks of safety case module matching and composition. It is expected that use of controlled language and terminology (i.e. CCL developed by WP4) within the statements of argument structures could be beneficial for parsing of the structures and would provide the basis for development of some assistance tools. The authors are not aware of any pre-existing research in this area.

8.2.2 Implicit context and assumptions in safety case evidence

In addition to the challenges of managing (and matching) explicitly declared contexts, it should be noted that some of the important contextual information may be hidden within the evidence cited by safety case modules. A trivial example is a question of whether the evidence presented in different modules relates to the same (or compatible) version of the system, configuration and set of requirements.

More critically, virtually all safety analysis methods, in practice, require safety engineers to make a number of simplifying assumptions. Some of those are ‘embedded’ in (or heavily influenced by) the analysis methods themselves, whereas others are a ‘free choice’ of safety engineers. One example is independence assumptions that are routinely made in FTA (such as independence of power sources for redundant aircraft system channels). Independence assumptions are not embedded in FTA (although the method is heavily biased towards them). In contrast, “once failed always failed” and binary simplification (a fault is either
present or not) are examples of inherent assumptions that are forced by the analysis method onto safety analysts. Either type of assumptions will not necessarily be explicitly declared in the safety case and may be hidden inside the cited evidence.

Clearly, safety case modules can only be justifiably composed if assumptions (and other contextual information) associated with evidence cited by different modules are compatible. Whilst some work has been previously done on systematic recording of the safety analysis assumptions\(^8\), we are not aware of any work that can be readily reused or easily adapted for the context of modular certification in general. The problem of implicit contextual information can be partly addressed by disciplined recording of assumptions during analysis. It is expected that appropriate format for recording of this information will be developed jointly with WP6 and WP4 of OPENCOSS. Furthermore, the taxonomy of safety case evidence, presented in OPENCOSS D6.1, can be used as a starting point for documenting assumptions inherent in different types of evidence as well as for structuring the guidance for selected evidence types.

### 8.2.3 Reflecting modular structure in confidence and compliance arguments

In general, within the safety assurance research community, safety cases are increasingly viewed as consisting of three types of arguments:

- **risk** (or “primary”) arguments – that aim to establish that the system is acceptably safe to be deployed
- **confidence** (or “backing”) arguments – that are used to justify that sufficient confidence can be placed in evidence and inferences of the risk arguments
- **compliance** arguments – that show that requirements of the applicable standards have been satisfied

These three aspects of safety cases – sometimes seen as analogous to the concept of “view-types” in software architecture community – are interlinked. However, research and evaluation work on compositional/modular safety cases has until now focused almost exclusively on risk arguments. One of the challenges that should be addressed by OPENCOSS project is reconciliation of modular safety case structure with the above risk-confidence-compliance ‘paradigm’. This challenge will require co-ordinated response by WP4, WP5, WP6 and WP7 of the project. The scope of primary responsibilities of each of those work packages in this regard has to be clarified.

### 8.2.4 Trade-offs between compositional certification and monolithic safety case development

If safety case modules are developed fully independently with no coordination at the onset, it is highly unlikely that an adequate safety case for an integrated platform can be compiled in a ‘bottom-up’ fashion. A more realistic process model would require a safety case architecture to be established first, to identify key responsibilities of individual modules. For sufficiently complex safety-critical systems this in itself is a complex engineering task.

---

\(^8\) For instance MISSA project (see section 10.1) has established a format and developed guidance for recording of safety analysis assumptions in the context of application of model-based safety assessment (MBSA) techniques to analysis of civil aircraft systems.
Consequently, one of the pragmatic challenges faced by the compositional safety case development is initial “start-up” costs associated with establishing the overarching architecture. It is possible that for some classes of systems and/or for some features of system development processes and supply chains, these initial efforts and costs will outweigh the benefits of compositional certification. The OPENCOSS project should seek to obtain better understanding of the key parameters of these trade-off decisions and, if possible, to provide guidance for the users of the OPENCOSS platform.

8.2.5 Industrial evaluation

Finally, it should be noted that, whilst both the notion of safety cases and the application of structured argumentation techniques (such as Goal Structuring Notation) to safety case representation and development are well established and have reached industrial maturity, the concepts of modular and compositional safety case development are still an area of active research.

The basic principles of safety case modules and contracts were proposed over a decade ago (Kelly, 2001). However, we are aware of only one comprehensive and industrial-scale evaluation initiative of this approach to date – the Industrial Avionics Working Group (see section 5.4). On the one hand this can be seen as an opportunity for OPENCOSS project in terms of advancement of cutting-edge research. However, on the other hand, it poses significant challenges (in terms of the methodology reaching the level of maturity necessary for full-scale industrial application by the end of the project), since there is a lack of pre-existing feedback from practical application, and relatively little understanding of pragmatic challenges that arise from such application. This significantly increases the risks and uncertainty inherent in the endeavours of WP5 of the project.

8.3 Next Steps

This report is an intermediate deliverable of task T5.1 of WP5. The ultimate objective of the task is to elicit a set of requirements for the work package as a whole (these will be reported in the final task deliverable – D5.2). To achieve this, over the period of three months following publication of the present document the task partners will:

- Perform an in-depth review of the key relevant research initiatives that were identified in this document
- Clarify the scope and agree upon priorities for the work package
- Harmonise WP5 scope with the scope and requirements for WP4, WP6 and WP7
- Elicit, through analysis of WP2 outcomes and discussions within the WP5 team, a set of detailed requirements for the work package
Works Cited

ARINC. (n.d.). 653 Avionics Application Software Standard Interface Parts 1-3. ARINC.


Parnas, D. (1979). On the criteria to be used in decomposing systems into modules. In Classics in software engineering (pp. 139—150). Yourdon press.


Vardanega, T. (2009). Property Preservation and Composition with Guarantees: from ASSERT to CHESS. *ISORC.*


9 Appendix A

D.1. What percentage of the components of your products/applications are reused?
*Note: This includes all types of reuse and not necessarily cross-domain.*

D.2. What are the pros and cons of re-using components of products/applications in your specific work?

D.3. Is it feasible for you to reuse components together with evidence and/or safety arguments? Motivate your answer and indicate your percentage of success (a rough estimate) in this task.

D.4. What is the greatest challenge your organization currently faces concerning reuse of components together with evidence and/or safety arguments?

D.5. Do you have research or industry experience with Compositional, Modular or Contract-based (CMC) certification
   - Yes – please proceed D.6
   - No – please proceed to D.8

D.6. Please provide references to CMC approaches and technologies you find relevant to OPENCOSS

D.7. Based on your background in CMC certification, what can we and what can we not expect CMC certification to solve? Please briefly describe your success and failure stories.

D.8. What are the main motivators for using CMC certification in your organization?
   - Reducing certification costs
   - Making certification more systematic and manageable
   - Better analysis of system integration issues (emergent properties)
   - Finding markets outside the domain of your application for the systems you develop
   - Benefitting from relevant systems developed and certified outside your application domain
   - Others – please specify:

D.9. Based on your answer to D.8, which motivator is the most important for your organization?

Table 4: Questions related to WP5 in the survey that has been conducted with OPENCOSS partners
10 Appendix B

10.1 FP Research Projects

This appendix presents a set of initiatives related to WP5 (compositional certification, modular certification, contract-based certification and component-based certification), indicated by OPENCOSS partners. For each initiative, its purpose, the common characteristics with OPENCOSS and WP5, and the contribution of WP5 beyond the considered initiative are summarised.

The projects and initiatives have been divided into four categories: FP projects, ARTEMIS projects, ITEA projects and national projects. For a complete overview of projects that were taken into account for OPENCOSS, we refer to OPENCOSS D6.1.

ASSERT

The Automated proof-based System and Software Engineering for Real-Time systems (ASSERT project) was a FP6 European Project in Embedded Systems (April 2004 – December 2007). ASSERT focused on aerospace application domain and European Space Agency (ESA) leads the project. The main objective of the project was to introduce a new methodology to aerospace system development, based on modeling and on preservation of system properties from design down to the code.

ASSERT introduced two main design processes and supporting tools: one exploited Architecture Analysis and Design Language (AADL) standard language (Architecture Analysis and Design Language) and another one exploited RCM (Ravenscar Computational Model) (Vardagena, 2006), which is the metamodel of the Ravenscar profile, a tailored ADA profile to high-integrity real time embedded systems.

Some results of the ASSERT project have been successfully reused in other application domains via the CHESS ARTEMIS project (ASSERT project) (Vardanega, 2009). The industrial and academic feedback is positive and the some ASSERT results are still today relevant in the literature, and potentially useful within the OPENCOSS project.

First of all, the preservation of properties under the composition is provided by a set of techniques that exploit: (1) a contract-based interface approach to design (Cancila, Passerone, Vardanega, & Panunzio, 2010); (2) a separation of concerns between functional and non-functional properties (Vardagena, 2006) (Cancila, Passerone, Vardanega, & Panunzio, 2010), and; (3) an interface grammar that supports model transformations (Cancila & Vardanega, RCM Interface Grammar) – thus proving composability (Sifakis, 2005).

Moreover, the ASSERT project ensures a current structure of a system model, i.e., safe task interaction behavior, by avoiding infinite call cycles, and it ensures satisfaction of timing properties (deadline, period, etc..) (Cancila, Passerone, Vardanega, & Panunzio, 2010) (Panunzio & T. Vardanega, 2009). This result provides an excellent basis for certification: a model is correct from design to code, the model supports composition and separation of concerns, and the model should conform to safety norms and, hence, be
Finally, the OPENCOSS project could adapt the successfully tested contract-based approach to design to certification issues.

MAENAD
This three-year project (MAENAD project), start-date September 2010, wants to deal with the new complex power management and optimization algorithms that are needed to ensure high performance, range of travel and low energy consumption to Fully Electric Vehicles (FEV). The challenges faced in the engineering of FEV are already partly met by EAST-ADL2, an emerging automotive architecture description language (ADL) compliant with AUTOSAR, and EAST-ADL2 is the appropriate vehicle for fully meeting these challenges. MAENAD will extend EAST-ADL2 with advanced capabilities to facilitate development of dependable, efficient and affordable FEV. The project will achieve language and tool support for:

- Support for the ISO 26262 automotive safety standard, including a novel approach for automatic allocation of safety requirements to components of an evolving architecture
- Effective model-based prediction of quality attributes of FEV such as the dependability and performance, via use of advanced, scalable, automated techniques.
- Automated exploration of potentially huge design spaces to achieve better or optimal trade-offs among dependability, performance and cost.

The scope of the modeling language and analysis focuses on the system structure and dynamics, in terms of physical, computational and communication components, their composition and interactions.

The link to WP5 could be found in the concept of an evolving architecture, but the work is still in progress.

MISSA
More Integrated Systems Safety Assessment (MISSA project) was an FP7-funded collaborative project that has run from April 2008 until June 2011. The project was funded under the Transportation (AAT) theme of the Framework 7 programme and was specifically concerned with the safety assessment of civil aircraft and its systems. The consortium comprised of 13 partners coordinated by Airbus Operations Ltd (UK); of those partners Atego UK, Thales and The University of York are also participants of the OPENCOSS consortium. MISSA was predominantly a follow-up from earlier ESACS (FP5) and ISAAC (FP6) projects. The core of the project was concerned with application of the novel Model Based Safety Assessment techniques for assessment of aircraft systems (consistent with the broad requirements and the framework of ARP-4754 and ARP-4761 guidance documents). Whilst earlier projects had focussed on supporting Preliminary System Safety Analysis stage of the aviation safety assessment process, MISSA expanded the scope to include a Functional Hazard Assessment stage, and to consider multi-systems assessment and also the overall synthesis of analysis results (consistent with the requirements of Preliminary Aircraft Safety Assessment [PASA] activity introduced in the recent revision of ARP-4754a). The core work of the project was undertaken through four main work packages:

- WP3 – concerned with Functional Hazard Assessment, optimal allocation of safety requirements to systems as well as optimisation of allocation of applications/partitions to IMA hardware modules (CPIOMs)
- WP4 – concerned with the preliminary safety assessment at the architectural level of design on the basis of AltaRica Dataflow modules (instead of traditional Fault Tree Analysis). This work package
covered both single- and multiple- system assessment (corresponding to PSSA and PASA processes respectively)

- WP5 – concerned with the safety assessment of detailed design proposals (specified in SCADE and Matlab Simulink environments) and adaptation of hybrid model checking and time compression principles to the model checking environment originally developed in ESACS and ISAAC projects.

- WP6 – concerned with the consolidation of safety analysis evidence into an overall aircraft safety case as well as with justification of model and analysis adequacy (i.e. construction of the backing argument), systematic tracing of modelling assumptions that underlie the analysis and, on the tooling level, establishing data flow between different analysis activities (on the basis of ATEGO Workbench environment).

Of relevance to the present report is the work carried out in WP4 and WP6 of MISSA (coordinated by University of York and ATEGO respectively). Within WP4, approaches to composition of multiple system models were developed. These were based on the notions of dependent system failures and multiplexer virtual components. Virtual components allowed harmonising and, where necessary, extending the interfaces of independently defined system models. Whilst, in OPENCOSS terminology, the composition has been performed at the level of evidence rather than that of a safety case, the experiments performed by MISSA have demonstrated a very significant computational complexity burden for performing multi-system assessment even at the relatively abstract level of AltaRica models. This suggests that a composition approach based on a notion of modular safety case and systematic tracing of implicit and explicit system-level assumptions (to be discharged by safety case fragments specifically dedicated to system composition within the aircraft) may be beneficial from the industrial perspective.

Within WP6 of MISSA, the framework of ARP-4754 and ARP-4761 documents was translated into generic safety case architecture (Figure 19: Simple Safety Case Architecture Example), captured in the Goal Structuring Notation. The architecture captured interdependencies between various safety assessment activities (such as aircraft and system level FHA, PSSA, SSA and Common Cause / Common Mode Analysis). Safety Case patterns have been developed for safety assessment activities targeted by other work packages (i.e. predominantly – FHA and PSSA). These patterns have covered both the primary argument (concerned with arguing safety of the system on the basis of the model analyses) and the backing argument (concerned with justifying the adequacy of the models and, thus, trustworthiness of the primary safety case). Patterns were adapted specifically for the model-based techniques developed and used in MISSA. Finally, related to the backing argument patterns, a framework for systematically identifying and recording modelling assumptions was developed. Overall it is important to note that the focus of MISSA WP6 was not the composition of a safety case from modules related to different systems, components and equipment but rather the composition of modules related to safety assessment performed at different stages of the safety lifecycle promoted by the de-facto safety standards in the civil aviation sector.

MODSafe
This project (MODSafe project) aims to undertake research of major steps of the Safety Life Cycle of urban guided transport systems in Europe. Even if the rail safety landscape in urban guided transport is highly diversified, the sector will benefit from some kind of harmonization. Furthermore, security items are considered more and more as vital for the urban transport sector.
The MODSafe project successfully started in 2008 with state-of-the-art evaluations and initial models. Hazard analyses, safety requirements as well as functional and object models, have been developed in the safety sector, while a life cycle approach proposal and an approval approach were established in the process sector. For the security sector, the existing means and technologies for security systems have been analysed, forming the base for a model reference under development.

Final results focus on cross-acceptance of proven and certified technologies. These activities help to create common safety and security methods, in order to reduce barriers within the European Union. As a result, competition and common/equal safety standards may be enabled. MODSafe however also shows the limits of standardization for technical safety functions and objects, as the consensus building process has shown.

At the point at which the project closes in August 2012, we can expect reasonable suggestions for the future, aiming to contribute to the European drive to harmonize and to simplify the upgrade/modernization or new construction of urban guided transport systems. Cross-Acceptance is one of the key attempts to the benefit of all parties involved, be it the manufacturers and suppliers, the operators of the safety authorities.

Relevance for WP5 should be:
- MODSafe project analyses the state-of-the-art in safety, to identify how the safety approaches are used in a large number of European Member States;
- MODSafe also analyses certification and presents some approval processes applied in the European Union and Australia, making a comparison of current life cycle approaches in European Member States;
- MODSafe also presents a proposal of a common safety life cycle approach;
- Finally MODSafe analyses the different approaches in Acceptance, Approval and Certification, making the diversity transparent for participants of these processes.

Contribution for WP5 shall be to provide the state-of-the-art in safety and to provide a starting point to develop a compositional life cycle approach.

**SPEEDS**

The Speculative and Exploratory Design in System Engineering European project (SPEEDS project) is a FP6 European Project in Embedded Systems Development (May 2006- April 2010) The main objective of SPEEDS is to define end-to-end methodologies, processes and supporting tools for embedded-system design (SPEEDS white paper).

SPEEDS defines a heterogeneous rich-component (HRC) (meta)model and a mathematical structure that allow engineers to soundly compose heterogeneous subsystems (Passerone & all., 2009). A contract-based design supports such a heterogeneous composition.

A contract is a pair of assumptions and promises, where an assumption is an acceptable environment or context for a subsystem, and a promise is a possible behavior of that subsystem with respect to the context, declared in the corresponding assumption.
Contracts are associated with the subsystems and components of a system via ports, that is, special interaction points among subsystems (or components) and between a system and its subsystems. The information, which a contract describes, can be exploited not only to soundly compose heterogeneous components, but also to analyze, test and verify a system design. This is, for example, the case of safety-related information, such as safety goal (which OPENCOSS exploits in WP4) or safety requirement: it is specified by a contract and, then, associated to a component of a system and exploited to analyze the system with respect to safety issues.

A limitation of the SPEEDS project is not to be adherent to international standards, such as SysML. However, its mathematical structure and the model-based design methodology are still valid. OPENCOSS WP5 work can reuse the SPEEDS study from mathematical and model based-design approach viewpoints.

10.2 ARTEMIS Joint Undertaking projects

CESAR

This project (CESAR project) addresses industrial needs for embedded system development for safety relevant applications by developing ultra-reliable embedded components for use in an extremely competitive global market requiring drastic cost reductions. Such components are key, both for the societal needs of ensuring safety and mobility, as well as in providing “green products”, which facilitate the achievement of European targets for CO2 reduction. Applications developed in CESAR aim to address these societal needs, and to demonstrate the cross-sector relevance of CESAR innovations in five application domains: automotive, avionics, automation, rail and space.

The goal of the project is to introduce in each domain at least one significant innovation in the design, integration or validation process. Furthermore, it aims to ensure that those innovations are acceptable across the supply chain and, when appropriate, to certification authorities. The overall objective is reduction of development time or effort, between 30% and 50%, depending on the domain.

The achievement of the goal of the project is structured as following:

- Creating the European cross-sector standard reference technology platform (RTP) providing meta-models, methods, and tools for safety-relevant hard-real-time system development;
- Supporting holistic multi-criteria end-to-end design flows from system conception and requirement capturing to system realization based on a standardized formal requirement capturing language;
- Providing guidance for the optimization and assessment of systems/multi-systems architecture choices with respect to business and operational criteria (cost, safety, reliability, minimization of system interfaces, response times mass, ...);
- Providing complete encapsulation and full design reuse through multi-criteria rich component models;
- A suite of multi-criteria design, analysis and validation methods supporting consistency analysis, safety analysis, verification and validation supporting the CESAR reference technology platform.

The main contribution of WP5 with regard to this project is that safety issues regarding composability are taken into account. The CESAR initiative may be used to refer in some way directly to the composability criteria in general.
CHESs

This project (CHESs project), ending in April 2012, is partially funded by the Artemis Joint Undertaking. It sought to improve Model Driven Engineering practices and technologies to better address safety, reliability, performance, robustness and other extra-functional concerns while guaranteeing the correctness of component development and composition for embedded systems.

The CHESs methodology for the development of critical real-time systems is based upon a model-driven engineering approach, whereby the central artefacts of development are models rather than documents or code. It is a demanding approach that requires comprehensive tool support for all aspects of the development process.

In addition to being model based, the CHESs development process is component based, building on the important advances in component based engineering that have occurred in the last ten years. Special emphasis is set on providing advanced support for the rich and precise specification of the extra-functional characteristics of these components, especially in the dimensions of time predictability, isolation, transparency, dependability and safety.

The other distinguishing element of the CHESs methodology is the concept of separation of concerns. This concept is a long-established best practice for the successful development of complex systems, in which developers “divide and conquer” the problem by focusing separately on key aspects such as functionality, real-time behaviour, deployment – and even other stakeholder concerns such as economic costs – rather than by trying to capture all aspects simultaneously in one exceedingly complex big picture.

The methodological and technical means by which CHESs provides the developer with support for separation of concerns is the popular, well-accepted modeling concept of views, most prominently acknowledged in modern modeling languages such as UML. The CHESs Methodology associates a distinct view with a distinct concern pertinent to in system software modeling.

The CHESs development environment is based on extension plug-ins for MDT-Papyrus. MDT-Papyrus is an open-source modelling front-end with support for UML, SysML and MARTE. CHESs implements the principle of continuous verification that guarantees that the model is always consistent and up-to-date and that supports the iterative development. Incremental development is provided by its component-oriented approach, whereby systems are composed of components chosen from repositories. The CHESs tool-chain guarantees the implementation of the correct by construction paradigm by supporting:

- The analysis and verification of extra-functional properties (predictability, dependability and security) in the component-based software system modelling and assembly
- The propagation of the results back to the model to assure guarantee of the extra-functional properties for the component compositionality and composability
- The consistent generation of code and deployment on the target platform.

The CHESs Infrastructure is also developed to guarantee that the extra-functional properties asserted at model level are provably preserved at run time.
CHESS treats safety within the more general framework of dependability, since other characteristics such as availability are also considered. Nevertheless, special attention has been paid to the identification and precise specification of properties related to safety. These properties have been formalized within models and incorporated into their proper contexts within a number of well-known safety analysis techniques (e.g., FMEA and FTA).

MBAT
This project (MBAT project) started in 2011 and will finish in 2014. It deals with development, integration and delivery of high-quality products in the transportation domain, such as safety-related products including airplanes, cars and trains. More concretely, MBAT focuses on V&V of embedded systems.

MBAT will provide V&V technology in the form of a reference technology platform targeted at the production of high-quality and safe embedded systems at reduced costs. The platform will be based on model-based testing technologies combined with static analysis techniques. The project also aims at providing a new approach to use (and reuse) specially designed test and analysis models as basis for model-based V&V.

The approach proposed in MBAT for the reuse of models and artefacts might be useful and relevant for WP5. Since both projects will be executed in parallel, it seems logical to arrange joint workshops and meetings, and some work has already been done towards this aim. By exchanging knowledge and experience, both projects can benefit. Nonetheless, initial proposals and ideas have been presented in some research publications, which have to be reviewed to assess the real relevance of MBAT to WP5.

RECOMP
This project (RECOMP project) stands for Reduced Certification Costs Using Trusted Multi-core Platforms. The project started April 1st of 2010 and has duration of 36 months. The RECOMP research project intends to form a joint European task force contributing to the European Standard Reference Technology Platform for enabling cost-efficient certification and re-certification of safety-critical systems and mixed-criticality systems, i.e., systems containing safety-critical and non-safety-critical components. Applications addressed are automotive, aerospace, industrial control systems, and lifts and transportation systems.

RECOMP recognizes the fact that the increasing processing power of embedded systems is mainly provided by increasing the number of processing cores. The increased numbers of cores is a design challenge in the safety-critical area, as there are no established approaches to achieve certification. At the same time, there is an increased need for flexibility in the products in the safety-critical market. This need for flexibility puts new requirements on the customization and the upgradability of both the non-safety-critical and safety-critical parts. The difficulty with this is the large cost in both effort and money of the re-certification of the modified software. RECOMP will provide reference designs and platform architectures, together with the required design methods and tools, for achieving cost-effective certification and re-certification of mixed-criticality, component based, multicore systems. The aim of RECOMP is to define a European standard reference technology, supported by the European tool vendors participating in RECOMP.

The RECOMP European standard reference technology may provide insights into the customization and upgradability of safety-critical parts, for changes in modules within compositional systems. However, the standards from OPENCOSS are not used in RECOMP, and therefore it is uncertain whether such insights are
useful for OPENCOSS. The compositional modular certification requirements from OPENCOSS may be relevant for the automotive applications of the RECOMP project. While both projects end in 2014, exchange of outcomes from both projects may prove to be a challenge.

**SafeCer**
SafeCer (SafeCer project) is the composition of two Artemis projects: pSafeCer, a two-year project launched in April 2011, and nSafeCer, a three-year project launched in April 2012. The two projects share their targets to increase efficiency and reduce time-to-market of safety-relevant embedded systems by composable safety certification.

The primary SafeCer objectives are to provide support for system safety arguments based on arguments and properties of system components and to provide support for the generation of corresponding evidence in a similar compositional way. By efficient reuse during certification and stronger links between certification and development, higher degrees of component reuse is envisioned, as well as by providing support for reuse across domains the amount of components available for reuse will increase. SafeCer will provide a Tool Platform, where existing and developed tools interoperate to support the defined processes and a Certification Artifacts Repository to store and track the lifecycle of certification artifacts.

Sharing the same overall goals, the concepts developed in pSafeCer are advanced in nSafeCer into tangible industrial implementations of “project-ready”, unified and seamlessly integrated solutions, and demonstrators of the proof of concepts. The main industrial domains targeted in SafeCer are aerospace, automotive, construction equipment and railway. Other domains such as health care and cross-domain aspects will also be considered. Certification guidelines and training examples for various other domains will also be developed.

### 10.3 ITEA R&D Projects

**EVOLVE**
This project (EVOLVE project) started in 2008 and finished in 2011. The main objective of EVOLVE was the creation of a methodological framework for early V&V of evolutionary products (software for real-time embedded systems) through the accredited/certified integration of each iteration and component in a MDE context. EVOLVE provided an iterative and incremental methodological framework, driven by the agile and model driven development paradigms, fostering accredited/certified component reusability.

EVOLVE explored three tracks to add evolution and certification to MDE:
- Early V&V during the first phases of a project, based on models and requirements;
- Incremental and iterative model-driven development, and;
- Evolutionary certification on two levels: informal internal certification and homologation, and external official certification.

The major visible results expected for EVOLVE were:
- Unified modelling techniques to describe systems
- Iterative and incremental V&V techniques to develop certified components and products
- Certification activities integrated in normal development process
• Reuse of V&V and certification results and artefacts for new and derived variants of component and products

Although EVOLVE focused on V&V, it dealt with composition, through product development iterations.

10.4 Other initiatives

Flanders’ DRIVE ASIL (Automotive Safety Integrity Level) Project

This project (Flanders Drive project) has the goal to define an Automotive Safety Integrity Level (ASIL) and the development process the methodology got developed. Furthermore, not only was a methodology developed but also the document templates for each identified Work Product necessary for certification. FLAME covers the full lifetime of a product from early requirements gathering, over safety analysis, design, deployment and disposal.

While FLAME is targeted at the automotive domain, it still has some elements of cross domain certification in mind, because it also takes into account the safety requirements in the area of Off-Highway and Machinery, which bring in different requirements. During the analysis phase a lot of common requirements of the different standards were discovered, and documented.

The project itself is not concerned with compositional certification, but more with unification of the development processes to allow certification according to different standards, and even a little bit cross domain due to the inclusion of Automotive, Off-Highway, and Machinery standards to develop a common process flow.

The ASIL project is nearly finished, thus it is hard to contribute to the project, although Flanders’ Drive could be part of the industrial advisory board. Indirectly, contributions can be expected through the Altreonic partner in OPENCOSS. Altreonic has imported the FLAME process flow (renamed ASIL) in its GoedelWorks environment. In GoedelWorks, the user can develop his specific project from requirements capturing till implementation. A user organisation is deemed to customise and create an organisation specific version of the ASIL process flow and add their own processes when starting to develop a specific product or system.