

**Embedded multi-core systems for
mixed criticality applications
in dynamic and changeable real-time environments**

Project Acronym:

EMC²

Grant agreement no: 621429

| | | |
|----------------------------------|---|---|
| Deliverable no. and title | D12.1 – Requirements, specifications, and evaluation plan | |
| Work package | WP12 | LL_Cross Domain Applications |
| Task / Use Case | T12.1 | UC_Seismic surveying |
| | T12.2 | UC_Video surveillance for critical infrastructure |
| | T12.3 | UC_Medical imaging |
| | T12.4 | UC_Control applications for critical infrastructure |
| | T12.5 | UC_Railway applications |
| Subtasks involved | <ol style="list-style-type: none"> 1. ST12.1.1 UC_Requirements, specifications, and evaluation plan 2. ST12.2.1 UC_User requirements collection and synchronization 3. ST12.3.1 UC_Requirements, specifications, and evaluation plan 4. ST12.4.1 UC_Requirements and specification 5. ST12.5.1 UC_Requirements and specification | |
| Lead contractor | Infineon Technologies AG Dr. Werner Weber, werner.weber@infineon.com | |
| Deliverable responsible | Visure Solutions Almudena Díez, adiez@visuresolutions.com | |
| Version number | v1.0 | |
| Date | 19/12/2014 | |
| Status | Final | |
| Dissemination level | Public (PU) | |

Copyright: EMC2 Project Consortium, 2014

Authors

| Participant no. | Part. short name | Author name | Chapter(s) |
|------------------------|-------------------------|--------------------|----------------------------|
| 15N | Visure | Almudena Díez | Chapter 1, 7 |
| 12C | WG | Bjørn Nordmoen | Chapter 2 |
| 12A | Fornebu | Hans Petter Dahle | Chapter 2 |
| 01M | eVision | Michael Geissel | First Version of Chapter 3 |
| 04A | BUT | Michal Jurzykowski | Chapter 3 |
| 01U | Siemens | Jürgen Salecker | Chapter 3 |
| 15Q | ITI | Javier Cano | Chapter 3 |
| 02G | AIT | Axel Weissenfeld | Chapter 3 |
| 15L | HIB | Jesús del Peso | Chapter 3 |
| 17D | Philips | Mark van Helvoort | Chapter 4 |
| 17D | Philips | Bastijn Vissers | Chapter 4 |
| 17D | Philips | Elwin de Weerd | Chapter 4 |
| 17G | TU Delft | Imran Ashraf | Chapter 4 |
| 17H | Vector | Andrei Terechko | Chapter 4 |
| 16J | ABB | Tiberiu Seceleanu | Chapter 5 |
| 2J | TAT | Christoph Scherrer | Chapter 6 |
| 2J | TAT | Oscar Medina | Chapter 6 |

Document History

| Version | Date | Author name | Reason |
|---------|------------|---------------|---|
| v0.1 | 27/04/2014 | Alfred Hoess | Initial Template |
| v1.0 | 28/11/2014 | Almudena Díez | First version ready for review. This version is a compilation of the partial contributions to D12.1 from each WP12 use case (D12.1-N documents) |
| | 15/12/2014 | Almudena Díez | Version accepted after updates according to reviewer's comments. |
| | 19/12/2014 | Alfred Hoess | Final editing and formatting; deliverable submission |

Publishable Executive Summary

This deliverable contains the requirements of the use cases in WP12 in the ARTEMIS project EMC². These use cases come from different domains; the main objective shared by all of them is to take results from previous research projects and from the EMC² technological work packages and use them in innovative ways to prepare the ground for future multi-core applications.

These use cases are:

- UC12.1: Seismic surveying by ship
- UC12.2: Video surveillance for critical infrastructure
- UC12.3: Medical imaging
- UC12.4: Control applications for critical infrastructure
- UC12.5: Railway applications

For each use case a set of requirements and evaluation metrics are defined. Each requirement is described clearly and concisely; also a rationale behind each requirement is given, explaining the background to it. Each requirement is given a priority in order to be able to establish a ranking inside each use case and discern the most relevant aspects to be investigated. A verification method is defined for each requirement. Finally, a set of evaluation metrics is defined for each use case; they are quantitative and qualitative measurements that will be used in order to assess the demonstrators for each use case.

UC12.4 (Control applications for critical infrastructure) is about tool integration and is closely related with EMC² WP5; for this use case requirements are described following the template developed in the WP5 deliverable D5.1, using integration scenarios and user stories in order to identify the tools to be integrated and how they should be used.

Table of contents

| | |
|--|----|
| 1. Introduction | 9 |
| 1.1 Objective and scope of the document | 9 |
| 1.2 Structure of the deliverable report | 9 |
| 2. UC12.1 – Seismic surveying | 9 |
| 2.1 Description of the use case UC12.1 | 9 |
| 2.1.1 UC12.1 Objectives | 9 |
| 2.1.2 Description of work for UC12.1 | 10 |
| 2.2 Planned demonstrators in T12.1 | 11 |
| 2.3 Requirements | 12 |
| 2.4 Evaluation Metrics | 12 |
| 3. UC12.2 – Video surveillance for critical infrastructure | 12 |
| 3.1 Description of the use case UC 12.2 | 12 |
| 3.1.1 UC12.2 Objectives | 12 |
| 3.1.1.1 (UC 12.2.2) Tool instantiation for parallel processing | 12 |
| 3.1.1.2 (UC 12.2.3) Signal, image and video processing | 13 |
| 3.1.1.3 (UC 12.2.4) Passenger Tracking | 13 |
| 3.1.1.4 (UC 12.2.5) Access control and situational awareness | 13 |
| 3.1.1.5 (UC 12.2.6) Biometric access control system | 13 |
| 3.1.2 Description of work for UC12.2 | 14 |
| 3.1.2.1 (UC 12.2.2) Tool instantiation for parallel processing | 14 |
| 3.1.2.2 (UC 12.2.3) Signal, image and video processing | 15 |
| 3.1.2.3 (UC 12.2.4) Passenger Tracking | 16 |
| 3.1.2.4 (UC12.2.5) Access control and situational awareness | 18 |
| 3.1.2.5 (UC12.2.6) Biometric access control system | 19 |
| 3.2 Planned demonstrators | 21 |
| 3.3 Requirements | 21 |
| 3.3.1 UC 12.2.2 Tool instantiation for parallel processing | 21 |
| 3.3.2 UC 12.2.3 Signal, image and video processing | 22 |
| 3.3.3 UC 12.2.4 Passenger Tracking | 22 |
| 3.3.4 UC 12.2.5 Access control and situational awareness | 22 |
| 3.3.5 UC 12.2.6 Biometric access control system | 22 |
| 3.4 Evaluation metrics | 22 |
| 4. UC12.3 – Medical imaging | 23 |
| 4.1 Description of the use case | 23 |
| 4.1.1 Image processing | 23 |
| 4.1.2 Dynamic defect detection | 24 |
| 4.1.3 Runtime analysis of data communication and memory access | 24 |
| 4.2 Planned demonstrators | 24 |
| 4.2.1 Demonstrator Year 1 | 25 |
| 4.2.2 Demonstrator Year 2 | 25 |
| 4.2.3 Demonstrator Year 3 | 25 |
| 4.3 Requirements | 25 |
| 4.4 Evaluation Metrics | 25 |
| 5. UC12.4 – Control applications for critical infrastructure | 26 |
| 5.1 Description of the use case | 26 |

| | | |
|---------|--|----|
| 5.1.1 | Today's Procedures..... | 26 |
| 5.1.2 | Solution - Tool-chain realization | 27 |
| 5.1.2.1 | Application mapping on multi core systems | 28 |
| 5.2 | Planned demonstrators..... | 29 |
| 5.3 | Requirements..... | 29 |
| 5.3.1 | Tool-chain requirements | 29 |
| 5.3.2 | Pains & Benefits | 30 |
| 5.3.3 | User Stories..... | 30 |
| 5.4 | Evaluation Metrics | 30 |
| 6. | UC12.5 – Railway applications..... | 30 |
| 6.1 | Description of the use case | 30 |
| 6.1.1 | Fault tolerance services | 31 |
| 6.1.2 | Multi-core hardware platforms | 31 |
| 6.1.3 | Incremental certification and mixed criticality applications | 31 |
| 6.2 | Planned demonstrators..... | 31 |
| 6.2.1 | Safety Concept..... | 31 |
| 6.2.2 | Novel Programming Models | 31 |
| 6.2.3 | Virtualization with PikeOS | 32 |
| 6.3 | Requirements..... | 32 |
| 6.4 | Evaluation Metrics | 32 |
| 6.4.1 | Safety Case..... | 32 |
| 6.4.2 | Novel Programming Models | 32 |
| 6.4.3 | Virtualization with PikeOS | 32 |
| 7. | Conclusions | 33 |
| 8. | References | 34 |
| 9. | Abbreviations | 35 |

List of figures

| | |
|--|----|
| Figure 1: Seismic survey by ship | 10 |
| Figure 2: Biometric access control prototype | 14 |
| Figure 3: Biometric access control: web control panel | 14 |
| Figure 4: Application development for FPGAs | 15 |
| Figure 5: Typical object detection (face detection)..... | 16 |
| Figure 6: Passenger tracking | 17 |
| Figure 7: Experimental set-up of passenger tracking through various rooms. Passengers are tracked over various cameras and re-identified. | 17 |
| Figure 8: Expected evolution of HIB surveillance applications towards an access control and situational awareness application, based on EMC ² | 19 |
| Figure 9: "As is" (left) and "to be" (right)..... | 20 |
| Figure 10: Clinical workflow (typical example) | 24 |
| Figure 11: Processing and post-processing example (Smart DTI fiber tracking) [3] | 24 |
| Figure 12: Context for the Cooling System for Transmission Plant Application (CS_UC)..... | 26 |
| Figure 13: The original iFEST tool platform, showing the development focus points within EMC ² | 28 |
| Figure 14: Demonstrator: UC Specific Tools Framework. | 29 |

List of tables

Table 1: Quantitative metrics to be used in the assessment of results, for UC12.1..... 12
Table 2: Abbreviations..... 35

1. Introduction

1.1 Objective and scope of the document

The purpose of this document is to compile the requirements that should be fulfilled in order to achieve the objectives of the use cases in WP12, including both the ones that are going to be explored as part of the work in WP12 and the ones expected to be fulfilled by results found in the EMC² technological WPs (WP1-WP6).

In WP12 (Cross domain applications) there are five use cases from different domains:

- UC12.1: Seismic surveying by ship
- UC12.2: Video surveillance for critical infrastructure
- UC12.3: Medical imaging
- UC12.4: Control applications for critical infrastructure
- UC12.5: Railway applications

This document will be the reference for future work in these use cases and will also be shared with the rest of work packages in EMC² concerned by the requirements formulated in this document. Evaluation of results of the use cases will also take this document as reference for quantitative and qualitative assessment of the fulfilment of their objectives.

1.2 Structure of the deliverable report

The document is structured in five chapters, one for each use case. Each of them contains the following sections:

- Use case description.
- Description of the demonstrators planned in the use case.
- Requirements for the use case.
- Metrics to be used in order to assess the results of the use case.

The sections on use case description have been taken mostly from the Technical Annex, adding further clarification on the current status of technologies (AS-IS) and the expected situation at the end of EMC² (TO-BE), and some other minor refinements.

The requirements for each use case were written in table format in Excel files; these files are referenced in this document in order to preserve the readability of requirements.

The goal set with regard to the size of the deliverable was a maximum of 20 pages per use case.

2. UC12.1 – Seismic surveying

2.1 Description of the use case UC12.1

This chapter contains a brief description of the use case characteristics and objectives. The text below provides a description of the context and is fully compatible with the Technical Annex (v3.0 as of 6. Sep 2014), with some minor refinements.

2.1.1 UC12.1 Objectives

This task will establish a tool chain for seismic real-time processing on multi-cores. We want to (semi-) automatically generate low-level code, which can be executed efficiently, from the high-level formulation

of an algorithm. By doing so, this task will make a strong contribution towards two major, but interconnected, challenges:

1. Parallel Performance

For computations working to industrial standards, achieving excellent parallel computational performance involves a significant investment of time, as described below.

2. User-friendliness and Accessibility

User-friendliness is essential for the scientists formulating the algorithms, who do not have a complete knowledge spectrum covering numerical methods, real-time programming, high-performance programming, and hardware. Scientists want to develop new algorithms to test ideas for the improvement of results. A good testing environment should enable users to deploy their domain knowledge effectively while removing specifics that require knowledge from other domains, e.g., scientists are not knowledgeable about software engineering.

To meet these challenges, this task will

1. Design a new Domain Specific modelling Language (DSL) which targets seismic processing
2. Design and implement a prototype code generator, which targets embedded multicore architectures.

The results of this task will give European industry access to powerful tools for hardware that is available at highly-competitive costs.

2.1.2 Description of work for UC12.1

This industrial case from WesternGeco is about the signal processing and seismic processing which takes place on board of their seismic surveying ships. The processing on board the ship is a kind of pre-processing before the data are further processed at the large computing centres of Schlumberger. The on-ship processing requires several thousand cores to complete required computations within the allocated time boundaries. There are large amounts of data that must be processed within real-time constraints. Within 5 seconds of a 10 second measurement cycle, more than 2Gbyte of data must be processed.

Imagine an industry scientist who is used to writing code in MATLAB, making use of its convenient signal processing or matrix manipulation functionality. This scientist has a new idea for an algorithm that will overcome some problem with existing methods for a particularly large dataset. Testing the solution in MATLAB is not possible because of the parallel limitations, and spending several months translating the code to efficient C is a significant investment of both single developer time and the time of company developers without a proof-of-concept.



Figure 1: Seismic survey by ship

WesternGeco has a team of “algorithm designers” that use MATLAB¹ to formulate the best algorithms for a specific case of seismic processing. This team of “algorithm designers” typically consists of mathematicians and geophysicists. They experiment with various algorithms. MATLAB is the language of choice since it is the de-facto standard for formulation of these kinds of algorithms. MATLAB offers a wide variety of numerical algorithms, and it is easy to visualize the results of the computations.

¹ See <http://se.mathworks.com/products/matlab/> for more information about MATLAB.

When the “algorithm designers” have finished their job, then a team of programmers get the task of implementing the chosen algorithm so that it can execute efficiently on a set of multi-cores. It is possible to generate C code from MATLAB today, but this code is far from meeting the execution time requirements of WesternGeco. So, this is today a manual job, which requires expertise in writing low-level code that fully exploits the multi-cores, taking cache-size etc. into account. This kind of expertise is scarce, and the code produced is hard to understand, even for its authors.

To use tools that are available today to automatically generate e.g. C code from MATLAB and use parallel programming tools to distribute the C execution over the cores is not an option, since such generated code executes far too slowly. Instead WesternGeco transforms the MATLAB code by hand to Armadillo², an open-source C++ linear algebra library with syntax deliberately similar to MATLAB. Then the C++ code is tuned by hand to make the most efficient use of the hardware, taking its characteristics into considerations. As hardware is developing constantly, this way of working is costly. It is also costly because the tuning process has to be repeated when there are changes to the algorithm, which there almost always is. New architectures are becoming a more and more complex mix of multi-core and many-core components, which for this way of working is difficult and very costly to make use of.

The current industrial practice of having a team of modellers describing the algorithms and a team of programmers implementing these algorithms is not optimal. Having two separate teams is not efficient, there is much manual work in transforming the high-level models to low-level code, and it is costly, time-consuming and error-prone. It is easy to introduce errors when going from the high-level models to the low-level programming. Every change to the model requires a lot of low-level programming work. Multi-cores are notoriously difficult to program efficiently with today’s low-level languages, and a company gets very dependent upon the experts who write the code.

At the end of the project the current process for transforming specific numerical Matlab code into efficient parallel C++ shall be automated as much as possible. This shall be done without compromising the performance of the generated code.

The ideal solution would be to have a domain specific language for the kind of computations described above. It should be easy to describe parallel algorithms in a way which allows for efficient execution on multi-cores. Observation and measuring of the execution should be visualized at the language level, i.e. at an abstraction level similar to that of MATLAB. The language should be independent of the target, and it should be possible to execute the language on hardware without a proper operating system, e.g. in cases where you write your own scheduler. A pragmatic solution should focus on language constructs to support the industrial case of the project partners to reduce the scope. Many industrial partners have similar challenges, and the market for such a domain specific language is potentially large.

Reliability is a key factor for seismic surveying on the oceans, it is critical that the software is executing within the given real-time constraints and without errors.

2.2 Planned demonstrators in T12.1

At m20 (Nov 2015) the first prototype will demonstrate translation to C++, using the numerical code library named Armadillo. This code will have limited support for multi-cores. The prototype will be demonstrated in the first EMC² workshop held in or after m20. The purpose of the demonstrator is to verify that task T12.1 progresses according to plan.

At m33 (Dec 2016) the prototype code generator shall be complete, and the purpose of the m33 demonstrator is to show that the code generator fulfils the requirements of section 4.

² See <http://arma.sourceforge.net/> for more information about Armadillo.

2.3 Requirements

The approach followed in identifying requirements for use case UC12.1 has been to elicit what should be achieved from the person who knows the use case best: Bjørn Nordmoen. For this use case, no explicit requirements have been formulated to any of the technological WPs.

The UC12.1 requirements are included in the file Requirements_Document_WP12_UC12.1.xlsx.

2.4 Evaluation Metrics

Plan for evaluation of results are described in the table in Section 2.3.

The quantitative metrics to be used in the assessment of results are:

| |
|--|
| Time measurements of running the code generator. |
| Time measurements of running the generated code. |
| Calculate the cost of using the code generator at project end. |

Table 1: Quantitative metrics to be used in the assessment of results, for UC12.1

3. UC12.2 – Video surveillance for critical infrastructure

3.1 Description of the use case UC 12.2

This chapter contains a brief description of the use case characteristics and objectives. The text has been partially copied from the Technical Annex (v3.0 as of 6. Sep 2014).

3.1.1 UC12.2 Objectives

The main goal of this industrial case is to further develop and use computer vision technologies – or more precisely video streaming technology – to ensure that critical infrastructures are safe and reliable. The objectives are reflected in the five tasks of the use case that each will contribute to the main goal as well as to their own objectives.

- Investigation of tools for parallel processing and parallel environments.
- Research of implementation of signal, video, and image processing algorithms.
- Exploitation of algorithms of person tracking in safety and security applications.
- Identity detection and situation awareness in applications.
- Biometry and authentication systems applications.

These tasks can result in improving the integration of the algorithms in real-time hardware implementation of low level algorithms for video processing, hardware for higher level tasks, combined hardware and software solutions or industrial applications of better quality.

3.1.1.1 (UC 12.2.2) Tool instantiation for parallel processing

This use case will deal with populating algorithms from a high specification level into embedded video systems and develop technology to create market specific solutions on standard multi-core hardware platforms. They will do this by establishing integrated tool chains for HW&SW co-design and use them efficiently to reduce time-to-market and raise the achievable level of quality. The sub-objectives of the task include:

- Concept for simplification of the work process on a high level abstraction level and automated documentation creation in typical design processes.

- Integration of third vendor tool chains into the development environment and integration of novel tools.
- Processes for quality insurance of project in parallel and embedded environments.
- Integration of customer specific hardware into this new design flow concept.

3.1.1.2 (UC 12.2.3) Signal, image and video processing

The objectives of task T2 are motivated by the need to efficiently and safely exploit the existing large base of image and video processing algorithms and their implementations in the embedded systems, where both the requirements and resources are different from the standard platforms (PCs) where the algorithms are developed. The task is subdivided into activities corresponding to the following sub-objectives:

- Sensor interfacing for signal, image, and video – this objective includes research of methods of interfacing of signal, image, and video devices in the embedded system, data transfers, etc.;
- signal, image and video processing – this objective concerns general signal processing algorithms, image processing and video processing algorithms in the context of embedded systems and their resources plus relationships between the algorithms and the applications;
- legacy algorithms - exploitation of the existing algorithms and porting them into the embedded environment (e.g. Matlab, OpenCV, etc.).

3.1.1.3 (UC 12.2.4) Passenger Tracking

Transport infrastructures, especially airports, have a need for video tracking technology that works with the video generated by existing large networks of security cameras, to provide people monitoring and people flow prediction for better passenger handling. The main sub-objectives of the use case include:

- Requirement and specification collection for the tracking and similar video processing tasks.
- Exploit and validation of the EMC² architecture as a feasible computer architecture and specification of a system solution for the passenger tracking use case based on existing tracking algorithms.
- Based on the system specification, to develop and use a service oriented middleware for distributed CPU many-core and CPU/GPU multi-core computer architectures.

3.1.1.4 (UC 12.2.5) Access control and situational awareness

Access control and situational awareness of personnel inside a building has become a high demand both in private and public facilities. Existing products will be further enhanced and complemented in order to include the wide range of sensors installed in a building, jointly with fixed cameras imaging and any other type of real time input. The main objective is to be able to provide a complete system providing not only access control based on face recognition both from fixed and mobile devices, but also real time knowledge on inside situation of the building and decision support. The task is subdivided into the following sub-objectives:

- Requirement and specification collection for the situation awareness and identity detection and similar tasks.
- Building of a complete system providing access control based on face recognition in fixed and mobile devices.
- Real time system for building of knowledge of situation in buildings and situation awareness including decision support for the applications.

3.1.1.5 (UC 12.2.6) Biometric access control system

"**Face detection** for a Biometric access control system"

This sub use case is based on a **Biometric Access Control (B.A.C.)** system prototype, although the main target is to provide a **face detection algorithm** that will be integrated in a common demonstrator for all T12.2 partners. For the whole use case 12.2, an **execution hardware**, an **implementation flow**, and **different algorithms** will be used to create a multipurpose demonstrator of how to transfer algorithms implemented in current standard solutions to multicore embedded systems.

The Biometric access control prototype is a solution aimed to recognize people by means of facial image analysis algorithms. The pictures below show how a complete prototype device installed just before a building entrance can automatically perform the identification and access registration of every person who requires registering an exit or entrance action.

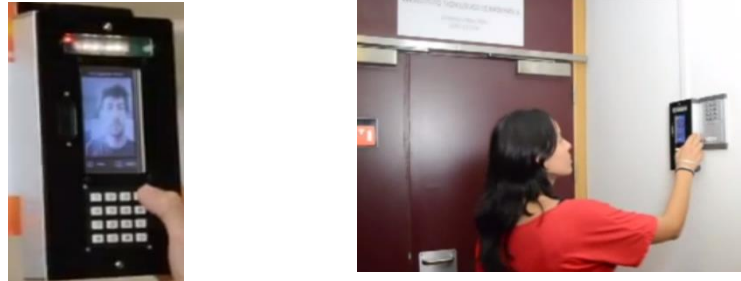


Figure 2: Biometric access control prototype

The implemented technology on this biometric access control system includes fast processing techniques, which allow for user identifications in less than 300 ms and large user databases. Next, a snapshot of the web control panel for access registration and user administration is shown:

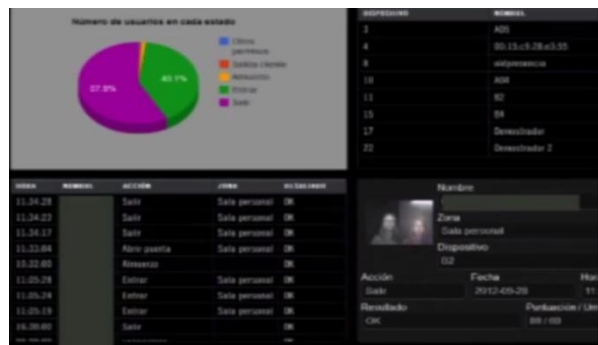


Figure 3: Biometric access control: web control panel

3.1.2 Description of work for UC12.2

3.1.2.1 (UC 12.2.2) Tool instantiation for parallel processing

Developing own technology for adding more value to our company is driving our activities in the last years. One major field of investment is video technology. Our goal in this project is to extend our expertise concerning populating algorithms from a high specification level into embedded video systems and develop technology to create market specific solutions on standard hardware platforms.

One – preliminary – result of the iFEST project is that industrial available tool chains for HW&SW co-design are not yet mature enough to be used in ordinary industrial context. The average developer will not be able to use those tool chains efficiently. However if those tool chains are in place and can be used efficiently the results are tremendous, not only concerning time-to-market issues but also the achievable level of quality will be significant higher as using less modern technologies.

The identified core problems are caused by: lack of proper documentation and training, gaps in the provided abstraction level, no or limited support for built-in quality insurance techniques and very limited support in case of custom embedded hardware needs to be included.

Setup of a design flow to implement examples of image processing into FPGA based boards. The concept is to implement system specifications on a high level of abstraction into dedicated hardware boards. The example project will use the implementation chains that are part of the project to verify the concept and to point out critical topics.

The use case will establish a custom specific Platform-Support-Package (PSP, green blocks, see figure below) for the selected tool chains. For the C-VHDL transformation commercial tool chains will be used and evaluated.

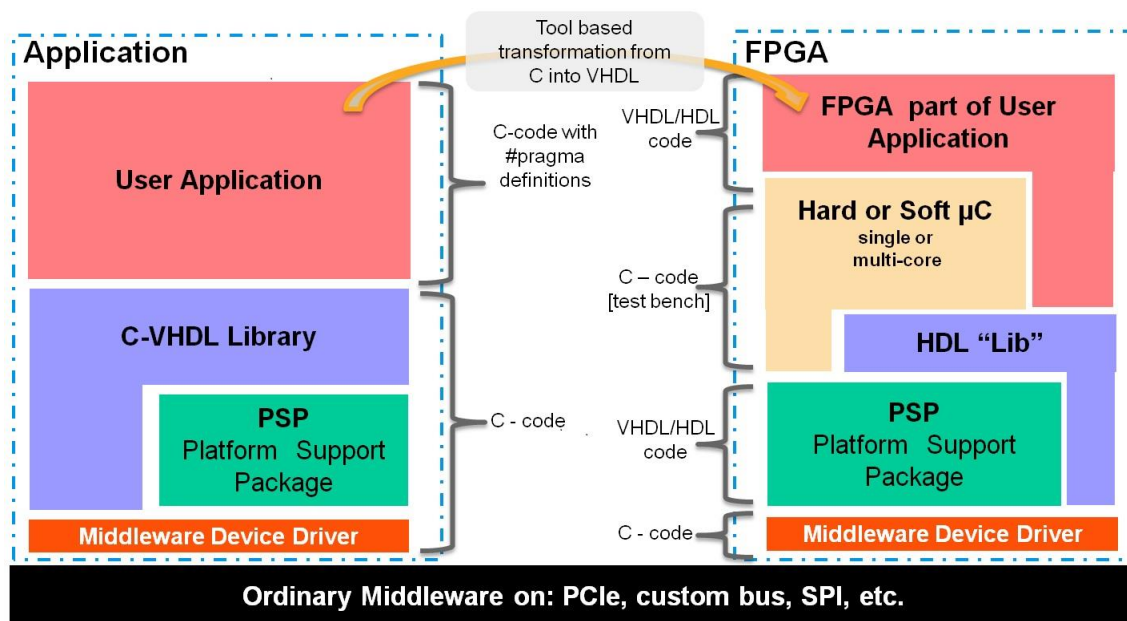


Figure 4: Application development for FPGAs

The proposed approach will make use of an FPGA which is NOT yet supported by the tool chain vendors. Main goals

- Concept for simplification of the work process on a high level abstraction level and automated documentation creation in typical design processes.
- Integration of third vendor tool chains into the development environment and integration of novel tools.
- Processes for quality insurance of project in parallel and embedded environments.
- Integration of customer specific hardware into this new design flow concept.

Expected results

- Practical – industrial valuable experience – how to work with a generic design flow and what is the effort required for such a custom project.
- Knowledge of how to use the tools for a generic development project and what is the required effort in such projects if non-standard PSPs have to be developed as well as the content of a professional training for those design flows.
- Influencing of the tool and hardware manufacturers to provide much better tool support.

3.1.2.2 (UC 12.2.3) Signal, image and video processing

The objectives of this subtask are motivated by the need to efficiently and safely exploit the existing large base of image and video processing algorithms and their implementations in the embedded systems, where both the requirements and resources are different from the standard platforms (PCs) where the algorithms are developed.

This subtask should be oriented towards methods of interfacing of signal, image, and video. However, the focus should be on data flow, handling of data in the embedded system, data transfers in multi-CPU and heterogeneous systems (CPU and FPGA, etc.), but also at protection of the embedded systems from data source failures. Moreover, the task should focus general signal processing algorithms, relationships between the existing algorithms and the application environments, effect of algorithms on the safety and security aspects of applications, and also adaptation of existing algorithms into the embedded systems of interest. The image and video processing will be addressed as well with focus on image processing and to a limited extent given by the capabilities of the embedded systems also to video processing. The actual goals of the task should be oriented towards the anticipated application areas of the project and will include e.g. object detection, tracking, etc. Finally, the task will address the issues of porting the existing algorithms of signal, image and video processing as well as exploitation of existing imaging and signal processing frameworks and applications in the embedded systems (e.g. Matlab, OpenCV, etc.) especially from the point of view of resource requirements, safety, and validation.

The addressed tasks can be illustrated e.g. through the following figure showing the typical object detection (face detection) in image task and its results.



Figure 5: Typical object detection (face detection)

Main goals

- Sensor interfacing for signal, image, and video – this objective includes research of methods of interfacing of signal, image, and video devices in the embedded system, data transfers, etc.
- Signal processing, image processing, and video processing – this objective concerns general signal processing algorithms, as well as image and video processing algorithms, relationships between the algorithms and the application environments, and effect of the algorithms on the safety and security.
- Legacy algorithms - exploitation of the existing algorithms and porting them into the embedded environment (e.g. Matlab, OpenCV, etc.)

Expected results

- Knowledge of the required algorithms for industrial purposes in EMC² as well as computational resource availability and other technological features.
- Implemented set of algorithms in the form of software and possibly also hardware configurations.

3.1.2.3 (UC 12.2.4) Passenger Tracking

AIT contributes to T12.2 by providing a use case “passenger tracking” on airports. Transport infrastructures, especially airports, have been identifying a need for video tracking technology that works

with the video generated by existing large networks of security cameras, to provide people monitoring and people flow prediction for better passenger handling.

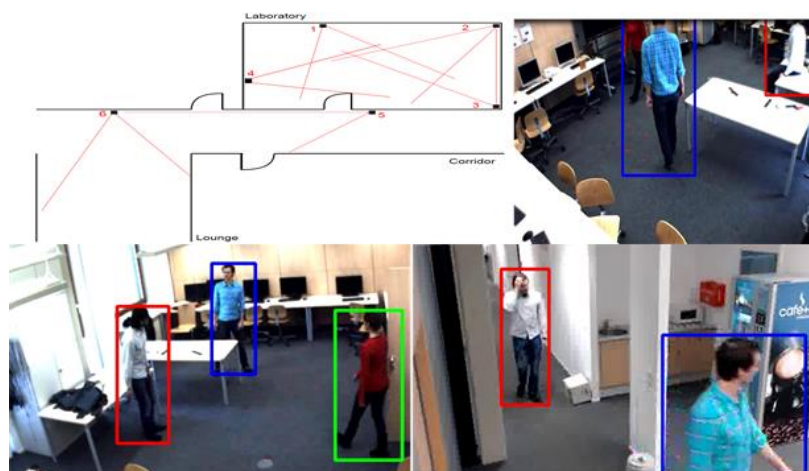
The figure shown below illustrates the visual tracking problem. IT solutions for this problem become a key technology for airport operators, however due to the difficulty of the problem a working solution is currently unavailable. AIT recently tackled this extremely challenging situation by proposing a new systemic approach to the visual tracking problem.



Figure 6: Passenger tracking

The upper row of the above figure illustrates tracking of a single passenger (red) in a security camera. Variation in viewpoint is large for different security cameras, hence tracking across cameras is an enormous challenge (bottom row). By courtesy of Flughafen Wien AG.

For this, an initial application has been developed to carry out a passenger tracking (Figure below). In that particular set-up six cameras are used for monitoring. The snapshots show tracked passengers in different camera views. Each passenger obtains its individual bounding box color. Same individuals even over various camera views receive the same bounding box color. The results underline that the application is capable to solve the use case in principle. In the real world, however, the use case is much more complex; for instance the number of cameras significantly increases as well as dynamic system changes – number of tracked passengers, lighting conditions, data bandwidth, computational resources, etc. We will solve those challenges by a middleware.



**Figure 7: Experimental set-up of passenger tracking through various rooms.
Passengers are tracked over various cameras and re-identified.**

The next step is to develop this middleware, which satisfies among others the following requirements: Automatic reconfiguration of the software/ hardware system is the main requirement to deal with dynamic changes of the environment. This requirement connects to the EU EPiCS project, which focuses on concepts and foundations for self-aware and self-expressive systems with novel hardware/software platform technologies and architectures for engineering autonomic compute nodes and networks. Results of the EPiCS project are incorporated into this use-case. The middleware which carries out system reconfigurations independent from the executed application shall not only support this particular use-case but rather provide functionalities (e.g., [1]) for sensor networks [2].

In this way, the system is adapted to dynamic changes (e.g., various data transfers from sensors) and the application capable to perform its tasks. In particular, computer vision applications (but also applications based on other sensors) benefit from this system behavior. For instance, a different type of tracking algorithm shall be executed depending on the number of visible passengers. Suitable hardware units (e.g., SoCs with multicore CPUs with GPU) are essential to provide sufficient computational resources for applications. The automatic reconfiguration shall enable the system for the efficient usage of available hardware units, data exchange between different hardware units, and handover of tasks from one hardware unit to the next as well as graceful degradation in overload situations.

Main goals

- Reduce the number of processors in network of cameras.
- Develop a service oriented middleware for passenger tracking in crowded areas like airports.

Expected results

- Service oriented middleware solution for networks of cameras with distributed computational nodes.
- Implementation and demonstration of the tracking system in the context of EMC² hardware and requirements.
- Exploitation of the service oriented middleware, especially communication services, in T11.2 for multimedia data.

3.1.2.4 (UC12.2.5) Access control and situational awareness

Access control and situational awareness of personnel inside a building has become a high demand both in private and public facilities. This use case will be directly covering the needs arising from private entities in order to safeguard the access to restricted buildings and at the same time have real time knowledge of what is the situation in their premises.

HIB is already offering to the market a solution set for addressing this issue, both *LiveonLife* for real time video streaming from mobile with advanced face recognition and *irc2* for the cloud semantic based information processing and decision support system.

In the frame of UC T12.2 these two products will be further enhanced and complemented in order to include the wide range of sensors installed in a building, jointly with fixed cameras imaging and any other type of real time input.

The objective is to be able to provide a complete system providing not only access control based on face recognition both from fixed and mobile devices, but also real time knowledge on inside situation of the building and decision support.

Currently HIB products are based on user mobile devices, such as smartphones, as processing units sending the captured and preprocessed data to a central server where information is integrated and finally processed. Currently smart phones and other consumer mobile devices provide huge information processing capabilities, making possible leveraging a part of the data processing burdens to these kind of devices. This makes possible to reduce central server processing requirements and therefore increase the number of sensors that can be integrated into the system.

Similar processing capacity is needed for fixed embedded devices in order to make them able to integrate into HIB's architecture. The application of EMC² technologies in this task is directly linked to the high number of heterogeneous, multi-core and disparate embedded systems and sensors that need to be addressed and integrated into the system in a holistic way: while currently mobile devices are usually based on widespread common operating systems and platforms, such as Android, this does not happen in the case of embedded systems. Having not only a common operating system but also a common development platform is of crucial importance for HIB to efficiently develop its solutions over embedded systems. Specifically, the Android virtualization support planned in EMC² Work Package 3 will provide a key for the development of a uniform application, both for mobile and fixed embedded systems. This will make possible to dramatically reduce the costs involved in the development of HIB applications and make them feasible.

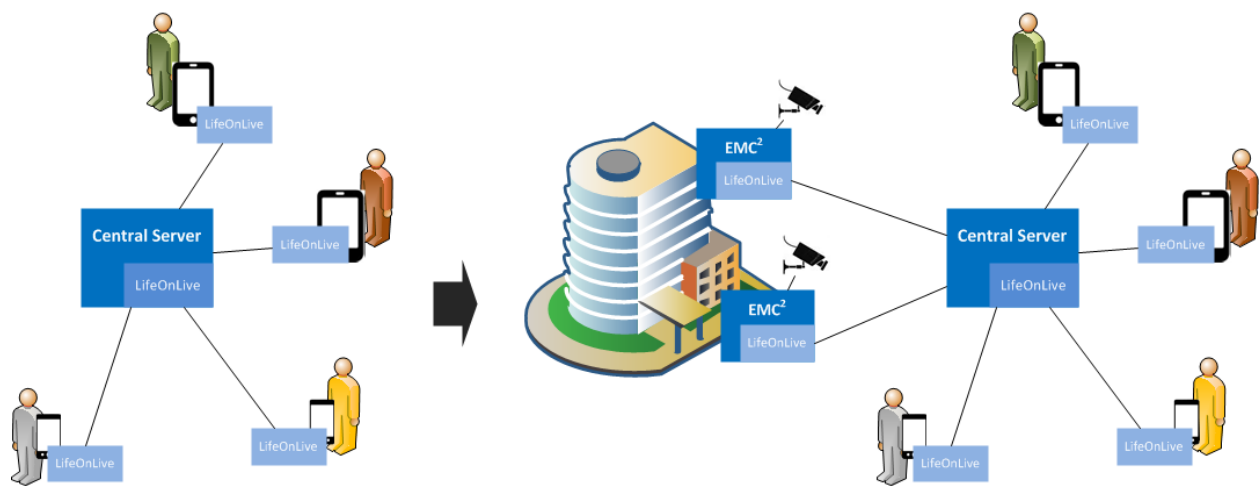


Figure 8: Expected evolution of HIB surveillance applications towards an access control and situational awareness application, based on EMC²

In addition, the real time requirements of situational awareness applications, especially for the situations that pose a risk for the individuals inside the building (i.e. risk of fire, structural damages, chemical or radiological threats, etc.) are expected to be also addressed by the EMC² multi-core multi-criticality platform.

The main advantages that HIB expect to obtain from the participation in EMC², for the *video surveillance for critical infrastructure* application, which will drive the evaluation of the use case can be the following:

- **Modularity, packaging and code reusability:** The use of the EMC² platform is expected to make that at least the 80% of the code and libraries originally designed and developed for mobile devices can be reused in EMC² based embedded systems.
- **Overall system performance increase:** Since a significant part of the image and video processing involved in video surveillance and situation awareness in fixed sensors is intended to be carried out into EMC² based embedded systems, a reduction of computation needs in central server is expected, which in other case would be in charge of the whole processing. Specifically at least of 50% reduction in computing load in central server is expected.
- **Cost reduction:** The code usability stated as first goal will have as a direct consequence the reduction of development costs for the integration of fixed embedded devices into HIB system. Therefore, an overall reduction of 25% in total development costs is expected.

3.1.2.5 (UC12.2.6) Biometric access control system

Each time embedded systems are used more and more in buildings. One of the most important uses of such embedded systems in buildings is related to grant access to the building (or a specific part of it). And

many different technologies are used in such systems for determining who can access and who cannot. ITI has developed a solution consisting on several intercoms with camera spread in all the access control points in the building that grant access based on face recognition.

The goal of this UC is to apply to this control system solution the architecture and paradigm associated to EMC² in order to decrease the computation requirements (and their cost) of the intercoms reducing their role to capture the face image. Moreover, all this process should provide real-time guarantees.

The main reason to have a distributed system for the biometric access control application is that acquisition hardware must be installed at every access point. Nevertheless, this hardware is not powerful enough to perform the heavy computation tasks required by the facial recognition algorithms, and it is not possible to use a high performance hardware (already available) or a dedicated one (specific development) due to its expensive cost in both cases. Therefore, a cloud model seems to be a convenient architecture to reduce production cost. In this task there will be a server which will be in charge of performing the main processing of the image, applying the face recognition algorithms and sending the corresponding action to the actuator (door lock) associated to the intercom that has sent the image.

The first work to perform in this task is to collect the requirements of the system and select an appropriate hardware suitable to perform some algorithms such as facial detection. Moreover this hardware must be optimized for power consumption and production cost. Additionally, SW architecture will be designed making use of the tools and methodologies proposed in EMC² for embedded software design, remote computation and real time systems. Biometric software and algorithms will be adapted to implement this methodology. The final result of this task is a client code application which integrates face detection in a low cost embedded system with remote computation. This implementation will be performed with generic and portable standards such as C standard language to allow multi-platform portable code.

Finally, validating the combination of HW and SW obtained in the previous task will be evaluated. Integration tests will play an important role and in this process especial emphasis will be put in ensuring the QoS of the obtained system.

Main goals

- Hardware design for distributed computing in biometrics embedded application.
- Design of SW architecture design in the embedded system.
- Development, integration, and testing of the biometric system.

Expected results

- Hardware architecture for biometry systems based on EMC² platforms.
- Implementation of the biometry system.
- Evaluation of the features of hardware, implementation, and practical exploitation of the system.

Current status of the face detection stage at the B.A.C.

The face detection stage of this biometric access control system is able to process 3 frames per second, running the algorithm on an ARM board based solution. The original algorithm employed to implement this stage has been coded in C/C++, making use of common features for every linux OS running on an ARM board like: large memory resources, advanced data types given by high level programming languages, etc.

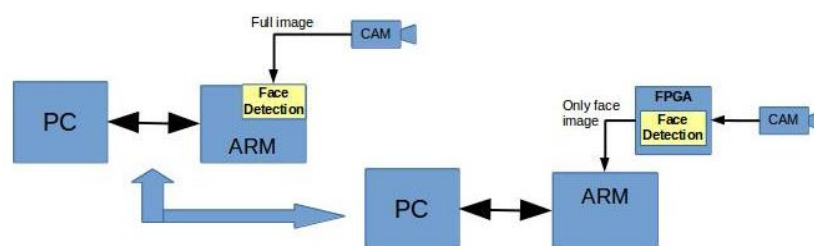


Figure 9: "As is" (left) and "to be" (right)

Aimed solution to improve the B.A.C. in the context of EMC2

The aim of this use case is to implement this face detection stage into an FPGA with a direct link with the camera. An adapted algorithm will be developed to obtaining a compatible version with the requirements imposed by the new execution hardware platform, i.e., a specific FPGA based board solution (to be defined in T12.2).

One of the main challenges to tackle in this use case will be to keep a tradeoff among processing speed, "goodness" of detection and hardware constraints. Nevertheless, the **main advantages** that could be obtained with this new approach are:

- **Performance increase** of the detection stage in terms of frame rate, raising the original 3fps to an expected 20fps.
- **Modularity and Packaging** of the B.A.C. system. The ARM board will be released of the face detection stage workload, and this may open the chance to a new, more flexible and cheaper, hardware architecture design for the biometric access control.
- **Cost reduction** of the B.A.C system given by the chance to avoid the use of high performance ARM board and using, instead, another lower-brand main board to control the door access interactive device.

A relevant task concerning EMC2 project is the need to **adapt the recognition code to meet FPGA standards** for the C programming language. Moreover, an **optimization of the source code** will also be needed prior to the migration to a FPGA **to reduce the use** of big image **memory maps**.

The collaboration with partners of subtask 12.2.6 will be needed in order to correctly perform the adaptation, optimization and migration tasks.

In this way FPGA requirements will be met by the code adaptation and optimization, whereas throughput will be raised by means of a devoted FPGA hardware algorithm implementation, clearly much faster than the current software implementation.

3.2 Planned demonstrators

After the first 12 month (April 2015) the first prototype will demonstrate a working HW platform to be used for FPGA based implementation of Video Algorithms. In addition the first implementations of video algorithms for the different use cases will be available.

At m33 (Dec 2016) the design process shall be complete and integrate the algorithmic work together with the design process. The purpose of the m33 demonstrator is to show that the example algorithms can be deployed to the proposed platform.

At month 20 (Nov 2015) the first prototype will demonstrate the capabilities of the middleware to foster the use-case "Passenger Tracking" by monitoring the system. The system will consist of a limited number of processing and sensor units. The middleware will be able to cope with some dynamic changes. The prototype will be demonstrated in the first EMC2 workshop held in or after month 20.

3.3 Requirements

A complete list of requirements can be found in the Excel file Requirements_Document_WP12_UC12.2.xlsx.

3.3.1 UC 12.2.2 Tool instantiation for parallel processing

Requirements for UC12.2.2 appear in the first lines of the Requirements_Document_WP12_UC12.2.xlsx Excel file. The UC Identification (first column) is WP12.2.2 (a). Some of these requirements are shared with UC12.2.3.

3.3.2 UC 12.2.3 Signal, image and video processing

Requirements for UC12.2.3 appear in the Requirements_Document_WP12_UC12.2.xlsx Excel file. The UC Identification (first column) is WP12.2.3 (b). Some of these requirements are shared with UC12.2.2.

3.3.3 UC 12.2.4 Passenger Tracking

Requirements for UC12.2.4 appear in the Requirements_Document_WP12_UC12.2.xlsx Excel file. The UC Identification (first column) is WP12.2.4 (c).

3.3.4 UC 12.2.5 Access control and situational awareness

Requirements for UC12.2.5 appear in the Requirements_Document_WP12_UC12.2.xlsx Excel file. The UC Identification (first column) is WP12.2.5 (D).

A major part of requirements comes from the need for a uniform execution platform both for mobile and fixed embedded systems. The requirements tackling this need are related to the work planned for several tasks in work packages 1 (Architecture) and 3 (Dynamic runtime environment and services). This involves for example, Android virtualization and support for standard and secured wireless communication protocols.

Other requirements are related to computing capabilities and performance to be supported by hardware and software subsystems in order to reduce processing load in HIB central server.

3.3.5 UC 12.2.6 Biometric access control system

Requirements for UC12.2.6 appear in the Requirements_Document_WP12_UC12.2.xlsx Excel file. The UC Identification (first column) is WP12.2.6 (E).

Some of the requirements come from the ITI experience on designing and building a biometric access control system prototype, as well as test experiences with real end users. **UC12.2.R0028** "Precision" is a requirement derived from the real test experiences, which tell us that a minimum "goodness" of the detection stage must be preserved in order to achieve a positive end user experience.

Requirements, like **UC12.2.R0033** "Meet FPGA programming language standards" and **UC12.2.6.e-R008** "Meet FPGA technical specifications" come from FPGA domain experts. A number of technical constraints must be satisfied by the algorithm in order to be migrated into a FPGA execution platform.

Requirement UC12.2-R007 from WP12.2 is affected by requirement UC12.2.R0030.

3.4 Evaluation metrics

Plan for evaluation of results are described in the Excel file in Section 3.3.

The quantitative metrics to be used in the assessment of results are:

- Improvement of performance:

12.2.6.e: Raise the throughput of the face detection algorithm from 3.5 fps up to 20 fps.

12.2.6.e: Reduce memory space required for internal computations from 35MB, to 710KB plus 500KB for the off-line trained data face detectors.

- Reduction of System Latency
- Reduction of Cost

4. UC12.3 – Medical imaging

4.1 Description of the use case

In the healthcare domain very large images need to be processed. Complex image processing is necessary to keep all image information available, while at the same time increasing the information level of the images. As very complex and configurable algorithms are in use, it is important to be able to develop the applications independent of the underlying hardware configuration, while assuring low latency for parts of the processing pipeline.

To prepare for evolution and to address the variability in products, it is necessary that the software can easily be ported to different underlying hardware configurations and the underlying heterogeneous hardware resources are transparently managed without application developers' intervention.

4.1.1 Image processing

Figure 10 shows a typical clinical workflow for creating a magnetic resonance imaging scan for a single patient. Based on the exam request received from the referring physician the radiologist determines in the morning which scan protocols should be used. Somewhere during the day the real MRI exam is planned, which takes 20 to 45 minutes. Only during this time-frame the patient is at the MRI scanner. Once the scan is done the patient is dismissed to either his room or even to home in case of an out-patient. Though some image processing is done during the scan time and a first evaluation is made by the operator, the real review of the examination data by the radiologist only takes place after the patient has left the scanning room. An example of processing and post-processing is given in Figure 11.

It is only at the moment when the radiologist is analyzing the data that it becomes clear whether or not the data is of sufficient quality to be able to perform a diagnosis on it. Even though the operator is performing a first check on image quality, it still requires expert eyes of the radiologist to qualify the final image quality. Also in the case of post-processing, which is performed after the exam has been conducted, the quality check is performed well after the patient has left the scanner (and likely the hospital). Problems in data acquisition may not be visible to the operator (which does basic checks on quality), yet may prevent the radiologist to draw conclusions from the post-processed image. If the data is of insufficient quality he has to order a rescan, which is both costly and puts an additional burden on the patient.

The solution to this problem would be to execute post-processing already at the MRI scanner. By performing more advanced checks that can be performed by the operator using post-processing, the quality check can be performed while the patient is in the scanner. If quality is proven to be insufficient, the scan can be done again. Multi-core computing is a pre-requisite for being able to perform such post-processing on the scanner console and the mixed criticality of such a system is the main hurdle to be taken with EMC² technology:

- Magnetic Resonance Imaging scanners create images based on spin physics. These physics dictate that latencies during scan and image pre-processing are very small (ns to μ s).
- The response of the scanner should give a real-time feel to the operator at all times (ms) and image processing should not cause timing problems for the very low latency activities.
- Post-processing currently can be done based on a time/cost-value base. When this process is run on scanner hardware, the requirement becomes similar to processing.

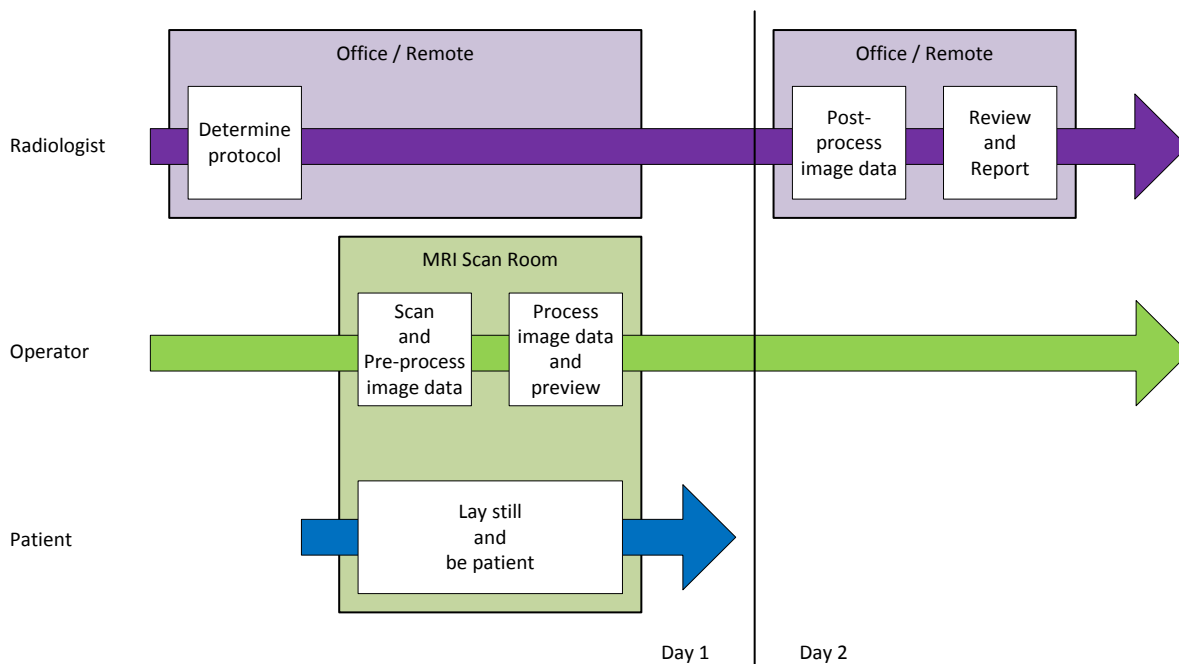


Figure 10: Clinical workflow (typical example)

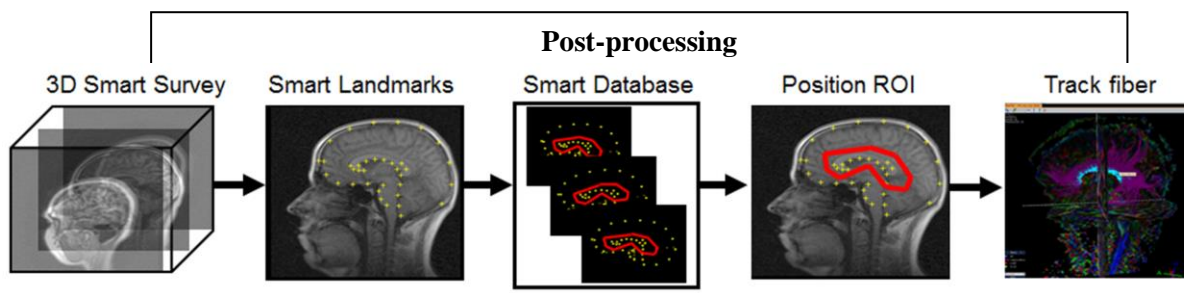


Figure 11: Processing and post-processing example (Smart DTI fiber tracking) [3]

4.1.2 Dynamic defect detection

The healthcare use case evolves from a large monolithic code base run on multiple computers towards a modular approach for application on a multi-core platform. From literature it is known that errors are part of large code bases and that new errors will be introduced when porting to a multi-core environment.

Instead of just analyzing the new source code statically, the source will be compiled into a special internal representation for analysis and executed in a virtual machine of the target architecture. This means the dynamic analysis engine has full visibility into everything that is going on inside the program. Based on the dynamic program analysis, the engine will report detected software defects.

4.1.3 Runtime analysis of data communication and memory access

To address the mixed criticality dynamic analysis tooling for optimizing deployment of algorithms on heterogeneous systems will be demonstrated. This tooling will map memory access of modules and identify communication channels between modules.

4.2 Planned demonstrators

The healthcare demonstration applies the EMC² technology in several healthcare situations involving medical images. The healthcare applications, served by the demonstrator, cover medical diagnostic and

intervention support systems. In interventional situations real-time processing is crucial. For efficiency reasons we need to be able to host several applications, with diverse latency and performance requirements on a single chip or board solution. This means co-location of real-time and non-real-time processing on a single chip/board. The algorithms that will run on the EMC² many-core execution platform involve massive image processing of different kinds (like noise reduction, image segmentation, image fusion etc.), but the background processing of the same piece of hardware deals amongst others with user interface, workflow support, decision support, administrative tasks, and (image) data storage and transfer. To ease portability, the demonstrators are built using domain-specific abstractions of the image processing algorithms.

4.2.1 Demonstrator Year 1

Setup of the full clinical workflow based on 5 distinct computers, with the following functions, where each function is deployed on its own computer:

- User interface
- Control
- Reconstruction
- Archiving
- Post-processing
- Viewing

4.2.2 Demonstrator Year 2

Setup of the clinical workflow with a merger of the user interface and reconstruction computer (which simplifies image data exchange):

- Scan (= User interface + Reconstruction)
- Control
- Archiving
- Post-processing
- Viewing

Dynamic defect detection and dynamic analysis demonstration of comparable code from the healthcare domain on a UNIX platform.

4.2.3 Demonstrator Year 3

Setup of the clinical workflow with post-processing and viewing of a highly demanding application on the scan computer:

- Scan + post-processing and viewing
- Control
- Archiving

Dynamic defect detection and dynamic analysis demonstration of healthcare code on a Windows platform.

4.3 Requirements

The requirements are given in the Requirements_Document_WP12_UC12.3.xlsx Excel document.

4.4 Evaluation Metrics

Plan for evaluation of results and quantitative metrics to be used in the assessment of results obtained by the use case:

- Post-processing and viewing of post-process is possible at the MRI scanner without hampering the low-latency processes and not disturbing the real-time control by operators. The demonstrator shall use a high-demanding application like DTI fiber tracking (see Figure 11) or compressed sensing.
- The dynamic analysis tool will run on a Windows platform and successfully detect software defects.
- The runtime analysis tool will enable high level partitioning decisions taking communication overhead in to account.

5. UC12.4 – Control applications for critical infrastructure

5.1 Description of the use case

ABB will create a prototype application covering the business domain of power system control. The main target will be a “Cooling System for Transmission Plant” (CS_UC) - Figure 12. The application is a closed loop control system where a number of sensor elements and actuators are connected by various interfaces. The system performs relevant actions depending on the input signals, the internal system state, the configurable logic, and possibly on operator commands. The system is required to perform a variety of computation intensive operations, with very high real-time requirements, on data coming in concurrent streams.

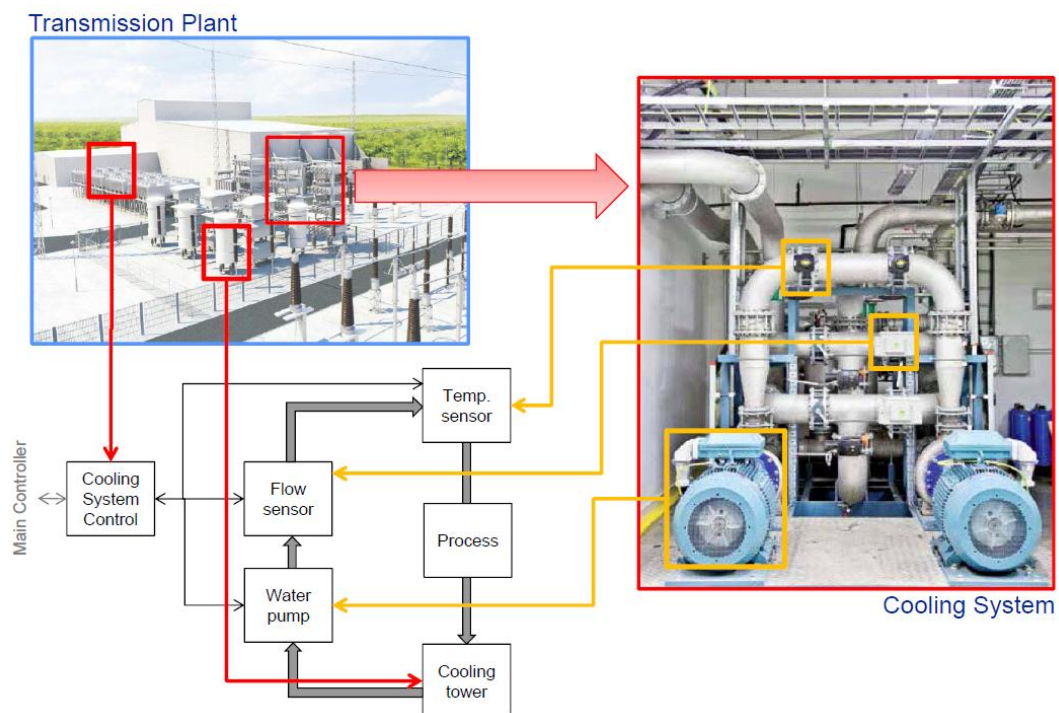


Figure 12: Context for the Cooling System for Transmission Plant Application (CS_UC).

The objectives here do not relate to the realization of the application as such, but focus on the creation and execution of a tool-chain to support the development, together with multi-core design aspects such as application partitioning and mapping. Thus, in brief, we target to (semi-) automatically provide an (almost) optimal solution for multicore SW/HW partitioning and mapping, and to (semi-) automatically exchange tools at different stages in the design flow within the used tool-chain.

5.1.1 Today's Procedures

There are several tools involved in the support for developing the CS_UC system.

Perhaps the most interesting current situation is that some phases of the development are not covered by tools with their main domain in the respective phase – such as Microsoft Word, used to collect requirements. This is one aspect leads to several problems as identified in the P&B form (see section 5.3.2). For example, the impact on effectiveness is qualified as: “Difficult to relate / manual relation to the other phases of the project(s), especially to V&V activities. Errors and omission to develop detailed and subsystem requirements.”

Another issue is the traceability of artifacts from inception (specifications) to implementation and further in the lifecycle phases. One of the reasons resides in the absence of connectivity between the used tools. There is no data produced by one tool that can be “somehow” inspected and analyzed by another tool, up or down the process flow. Hence, interpretations of data produced by (the users of) one tool at a different tool console (by another user) may be unclear, tedious and error prone. As a consequence, a long time may be required to balance all the viewpoints, for a correct interpretation and continuation of the development. Given the large number of (sub-)components of the CA_UC (which is a relatively small artifact in itself), this aspect has a large impact on the development time.

5.1.2 Solution - Tool-chain realization

The envisaged tool-chain is to be built on the basis established within the iFEST project [4][5], enabling tool replacement and tool independence. In addition, proper tools for specific tasks are also to be employed (such as Microsoft TFS / Visure Requirements for the requirements stages).

Several artifacts are to be developed within the project. Figure 13 presents a generic image of the iFEST tool framework, where tools are dynamically interconnected via tool adaptors - TA. The focus in the current project is to develop several such TAs for existing tools, but also to build, based on the project development (especially WP5 activities) additional tools.

At this moment, several tools have been identified as suitable to be used in the development of the application:

- Microsoft Team Foundation Server – to be used for requirement storage, code storage, collaborative artifacts. Tool adaptor to be built during EMC2.
- Visure Requirements – to be used for development and storage of requirements. Tool adaptor to be updated during EMC2.
- Enterprise Architect – to be used as an architectural development tool. Tool adaptor to be built during EMC2.
- HiDraw – a graphical design tool, used to build the application at various levels of abstraction (code to system levels). Tool adaptor to be built during EMC2.
- C compilers – multiple tools (depending on the target processor) for compiling the resulting code.

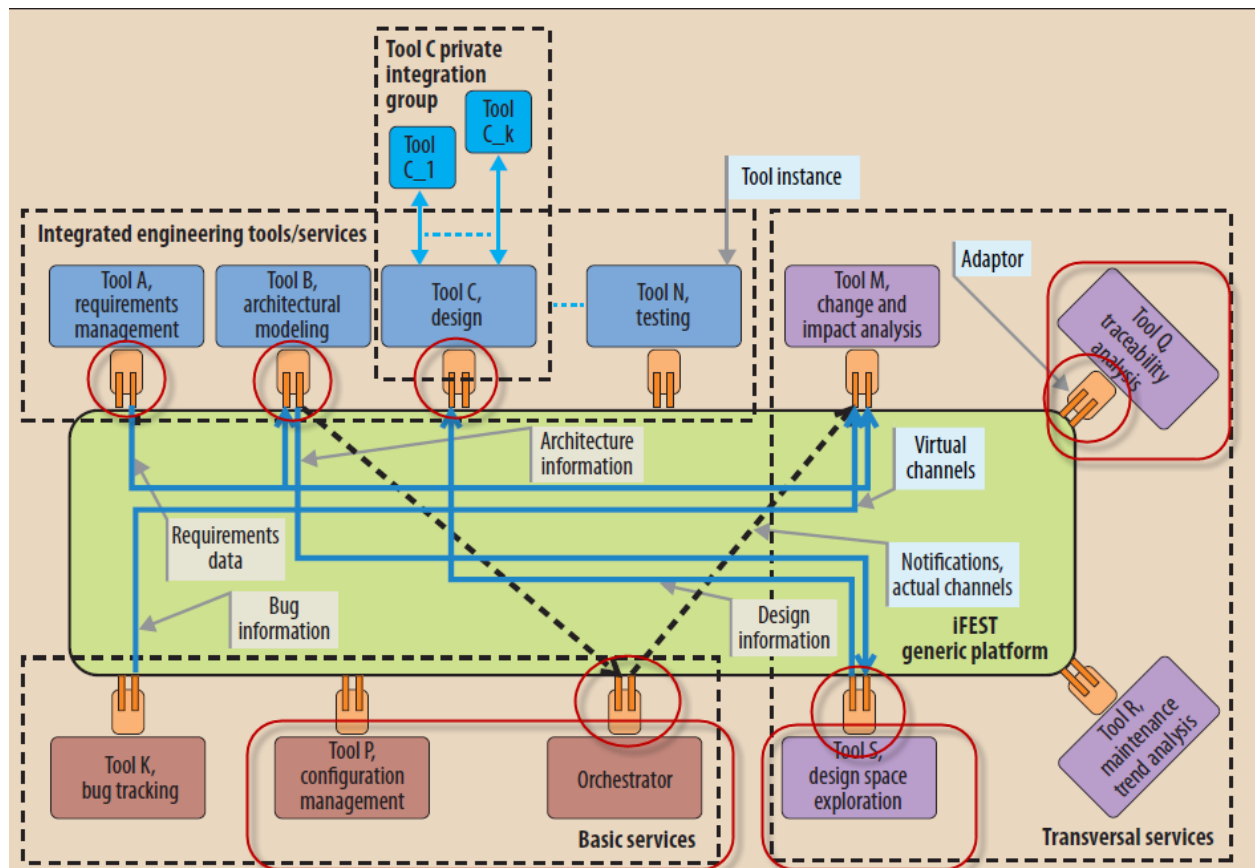


Figure 13: The original iFEST tool platform, showing the development focus points within EMC².

In addition to the above, new tools and their respective TAs are to be developed during the project.

- The **Orchestrator** will provide ways for transparent (source / destination independent) tool interaction.
- The **Multi-Core MultiPAR** tool [6][7]. This tool will complement existent solutions for design space exploration for (close to) optimal partitioning decisions on application mapping on multicore processors.
- A **Tool-chain configuration tool** is necessary for the seamless execution of the processes. This may eventually be part of the orchestrator.
- A **Traceability tool** will also be built in order to support the navigation through artifacts from incipient / requirement stages till late / implementation ones.

5.1.2.1 Application mapping on multi core systems

Typically, the system development is done in two stages. The basic platform components are developed first, and then a system is created with combinations of platform components and different user defined applications. The basic platforms consist of hardware (multi core CPUs, DSPs, FPGAs, ADCs, etc.), firmware and tools for engineering and configuration.

A permissive approach is targeted in the achievement of the first objective related to the mapping of the control application on HW / SW multicore technologies, allowing for a combination of library (existing component) and new designs. The decision on optimality of the solution will combine both technological aspects (HW / SW implementations), as well as process- and project- related aspects (available efforts, expertise, available / acceptable time-lines, etc.). The tool of choice here is MultiPAR.

5.2 Planned demonstrators

A single demonstrator will illustrate all the results obtained in the current task. The demonstrator will describe the development of the mentioned application with the support of an open tool chain addressing tool independence and multi core design space exploration.

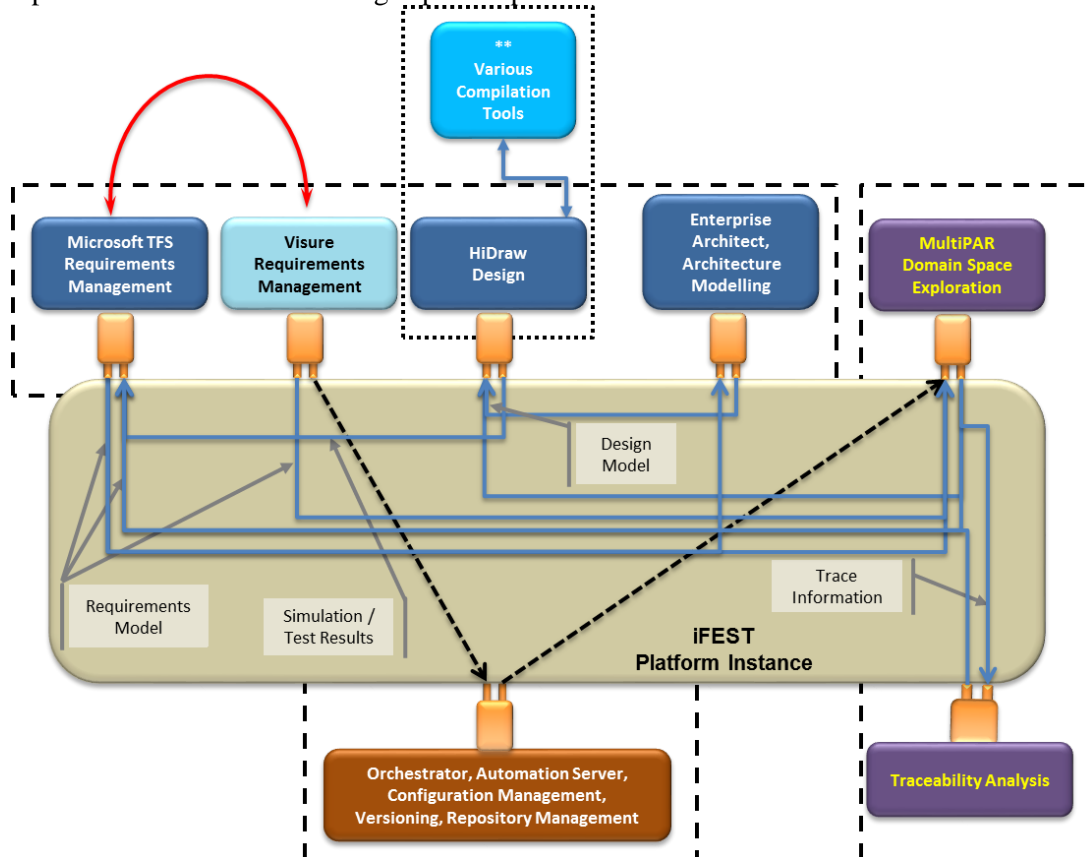


Figure 14: Demonstrator: UC Specific Tools Framework.

During the development of the application on the selected tool platform, two tools (*Microsoft TFS requirements* and *Visure Requirements*) will be exchanged to illustrate the idea of tool independence. The final result will benefit of the usage and decision support provided by the MultiPAR tool.

Subsequent goals of the demonstrator will be to provide figures on time to design, number of design errors, user accessibility and friendliness. These numbers will supposedly support the solution of the open tool-chain approach.

5.3 Requirements

As already mentioned, the purpose of the current use case is to build an open tool-chain to support an improved quality design of embedded systems and to provide a certain tool-independence.

As a consequence, the requirements are entirely focused on the tool-chain, with an approach based on the content described in deliverable D5.3 [8].

5.3.1 Tool-chain requirements

The set of requirements focused on the tool-chain realization are contained in the document *Requirements_Document_WP12_UC12.4.xlsx*.

5.3.2 Pains & Benefits

According to deliverable D5.3 there are two templates to fill in: (i) to highlight the issues met by designers in their operations within a given (tool-based or not) process flow; (ii) to both build specifications for building the integration artifacts foreseen by the tool framework to be used.

The so called “Pain & Benefits” (P&B) form helps the identification of issues within a given design flow. These are to be supposedly solved or somehow their non-beneficial impact should be minimized with the support of the intended tool framework.

For the current use case, the relevant P&B form is contained in the document QFD - Pains and benefits_UC12.4.xls.

5.3.3 User Stories

The next in the flow template to be used here is the “User Stories” template, or the “Tool Integration Matrix” (TIM) as described in [8].

The template tries to identify the building blocks for basically tool adaptor specifications, by descriptions showing how the tools within the framework are (planned) to be used.

The TIM corresponding to the current use case is given by the document WP12_ABB tool matrix.xlsm.

5.4 Evaluation Metrics

The main metrics of interest here can be identified as:

- Design development time
 - Intention is to have an as low as possible figure compared to current design times
 - Approximation of current design times will be provided
- Flexibility of the platform
 - To reflect the efforts to introduce or extract a tool from the chain
- User experiences
 - To reflect the benefits of users when working with the platform

6. UC12.5 – Railway applications

6.1 Description of the use case

This use case seeks to integrate technologies developed within EMC² into the roadmap of future generations of fault tolerant computer platforms for the railway domain. Thales’s TAS Platform will serve as a demonstrator for elaborated technologies with respect to

- Demonstration of safety in mixed criticality systems according to CENELEC standards using virtualization.
- Novel programming models for replica deterministic and parallel programming
- Safety concept for multi-core railway applications

The objective of this use case is to tackle challenges in the railway domain posed by multi-core architectures. The railway domain usually relies heavily on COTS hardware, which is steadily moving towards multi-core architectures. This trend demands new strategies for safety concepts, processing models and incremental certification for capitalizing on additional performance.

Thales Austria provides the TAS Control Platform [9] which is a fault tolerant computing platform that has been designed to meet the stringent safety requirements in the railway domain. It separates the railway-specific applications from the hardware and system software technology and serves as a common base for these applications, providing services such as time synchronization, membership service, voting, fault management and fault tolerance.

Thales Austria seeks to integrate technologies developed within EMC² into the roadmap of future generations of the TAS platform to overcome limitations imposed by the current state of the art. In particular:

6.1.1 Fault tolerance services

In safety critical systems that rely on interactive consistency, replica determinism of application execution is a crucial pre-requisite for fault tolerance services. In practice this mandates programming models that enforce a run-to-completion semantic. This paradigm, however to a certain extent contradicts parallel program execution, which inhibits taking advantage of modern multi-core hardware platforms. Therefore it is necessary to elaborate on programming models that allow parallelism without endangering replica determinism in a multi-core environment.

6.1.2 Multi-core hardware platforms

In the railway domain currently there is no established concept on how to demonstrate system safety for multi-core systems. Within this use case safety aspects of multi-core hardware technology will be scrutinized for single application scenarios as well as multiple and mixed criticality application scenarios.

6.1.3 Incremental certification and mixed criticality applications

In the railway domain a strategy to reduce the cost for recertification and to enable mixed criticality is to virtualize TAS Control Platform. Given that TAS Platform together with the application is virtualized a clear separation of computational resources can be established. This approach enables the coexistence of safety critical applications and non-safety critical applications on the same hardware cluster. In addition, careful resource allocation planning together with the creation of virtual TAS Platform clusters allow for later integration of independent applications.

Up to now virtualization approaches based on priority based scheduling algorithms have been evaluated during the course of the ARTEMIS nSafeCer project. These studies shall be extended to analyze time-partition approaches such as PikeOS as well.

6.2 Planned demonstrators

The planned demonstrators intend to visualize and showcase the research results gained in this use case. In particular three main topics are addressed:

- Safety concept for multi-core usage in the railway domain
- Novel Programming Paradigms
- Virtualization with PikeOS

6.2.1 Safety Concept

A core issue of actively redundant fault tolerant systems is the ability to test vital parts of the HW online during operation to avoid the accumulation of faults among replicated hardware. Such accumulation could impede majority voting and thus lead to positive vote on incorrect messages provided by replicas affected by accumulated faults.

The concepts for self-testing hardware in single core architectures are well established. However the introduction of multiple cores mandates enhanced algorithms, especially for testing system memory and caches. Care has to be taken to minimize the impact of memory testing in multi-core architectures on system reactivity. The demonstrator will analyze the impact of various algorithms and concepts of online hardware testing.

6.2.2 Novel Programming Models

The demonstrator concerning novel programming paradigms will show the results of the studies conducted to enhance the scalability of run-to-completion programming models on multi-core hardware platforms.

6.2.3 Virtualization with PikeOS

The associated demonstrator will serve as a test bed to study the impact of virtualization on TAS Platform performance with respect to jitter and predictability.

6.3 Requirements

The requirements associated with the Railway use case UC 12.5 (Thales AT only) can be found in the file Requirements_Document_WP12_UC12_5_3_p03.xlsx.

6.4 Evaluation Metrics

In the following the metrics and success criteria for the use case are defined.

6.4.1 Safety Case

The success criterion for this aspect is the certifiability of TAS Platform according to CENELC safety integrity level SIL 4 for multi-core architectures.

6.4.2 Novel Programming Models

Thales Austria seeks to improve performance by 20% when running TAS Platform on 2 cores instead of one.

6.4.3 Virtualization with PikeOS

The success criterion is the feasibility of the approach.

7. Conclusions

WP12 (Cross domain applications) consists of five use cases from different domains:

- UC12.1: Seismic surveying by ship
In this use case the main goal is to automate the current process for generating code that explores the computing potential of multi-cores.
- UC12.2: Video surveillance for critical infrastructure
The main goal of this industrial case is to further develop and use video streaming technology to ensure that critical infrastructures are safe and reliable.
- UC12.3: Medical imaging
The healthcare applications cover medical diagnostic and intervention support systems. This demonstrator deploys image processing algorithms using technology from WP2. Design tools may be used to expedite the engineering process depending upon the target product category. This demonstrator will illustrate the deployment of the algorithmic, execution platform and design tools, system software stack, contributions of the EMC² project in the healthcare use case.
- UC12.4: Control applications for critical infrastructure
In this use case a prototype application covering the business domain of power system control will be created. Nevertheless, the main objective focuses on the creation and execution of a tool-chain to support the development, together with multi-core design aspects such as application partitioning and mapping. The progress in use case 12.4 is highly dependent on the developments in WP5. The necessary orchestrator and tool adaptors are expected to be specified and implemented within the context of WP5 tasks. Specific measures for the objectives of interest will be collected in the next phases of WP12.
- UC12.5: Railway applications
The objective of this use case is to tackle challenges in the railway domain posed by multi-core architectures. The railway domain usually relies heavily on COTS hardware, which is steadily moving towards multi-core architectures. This trend demands new strategies for safety concepts, processing models, incremental certification and for capitalizing on additional performance.

D12.1 compiles the requirements that must be fulfilled in order to achieve the objectives of each use case in WP12, including the ones expected to be fulfilled by results found in other EMC² WPs.

Requirements are described as precisely as possible; not all of them can be quantified or defined in an accurate way but for all of them a verification method has been defined. Besides, for each use case quantitative and/or qualitative metrics have been defined in order to measure the degree of success achieved.

This document will be the reference for future work in WP12 use cases and will also be shared with the rest of work packages in EMC² concerned by the requirements formulated in the WP.

Evaluation of the demonstrators developed in each use case will take this document as reference for quantitative and qualitative assessment of the fulfilment of its objectives.

8. References

- [1] M. Garcia Valls, I. Lopez and L. Villar, "iLAND: An Enhanced Middleware for Real-Time Reconfiguration of Service Oriented Distributed Real-Time Systems," *Industrial Informatics, IEEE Transactions on*, vol.9, no.1, pp. 228-236, 02 2013.
- [2] B. Dieber, L. Esterle and B. Rinner, "Distributed resource-aware task assignment for complex monitoring scenarios in Visual Sensor Networks," in *Sixth International Conference on Distributed Smart Cameras (ICDSC)*, 2012.
- [3] Moerland, A., L. Geerts, H. Hoogduin, L. Moore, *Smart DTI fiber tracking through fully automated placement of regions of interest*, OHBM, 2013 (Artemis - High Profile result).
- [4] iFEST - industrial Framework for Embedded Systems Tools. ARTEMIS JU project #100203. Last access October 10, 2014, <http://www.artemis-ifest.eu/>.
- [5] T. Seceleanu, G. Sapienza, "A Tool Integration Framework for Sustainable Embedded Systems Development". *Computer*, vol.46, no. 11, pp. 68-71, Nov. 2013.
- [6] G. Sapienza, I. Crnkovic, T. Seceleanu, "Partitioning Decision Process for Embedded Hardware and Software Deployment, in proceeding of International Workshop on Industrial Experience in Embedded Systems Design, COMPSAC 2013. IEEE, Jul 2013.
- [7] G. Sapienza, T. Seceleanu, and I. Crnkovic. "Modelling for Hardware and Software Partitioning based on Multiple Properties. In 39th Euromicro Conference Series on Software Engineering and Advanced Applications. IEEE Computer Society, Sep 2013.
- [8] EMC2: Deliverable 5.3 – Integration requirements elicitation template.
- [9] Gerstinger Andreas, Heinz Kantz, and Christoph Scherrer. "TAS Control Platform: A Platform for Safety-Critical Railway Applications."

9. Abbreviations

| Abbreviation | Meaning |
|------------------|---|
| EMC ² | Embedded multi-core systems for mixed criticality applications in dynamic and changeable real-time environments |
| DSL | Domain Specific modelling Language |
| MATLAB | MATLAB® is a high-level language and interactive environment for numerical computation, visualization, and programming |
| Armadillo | Armadillo is a high quality C++ linear algebra library, aiming towards a good balance between speed and ease of use; the syntax (API) is deliberately similar to Matlab |
| DTI | Diffusion Tensor Imaging |

Table 2: Abbreviations