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	**D7.1 – Automotive use case descriptions, requirements and evaluation plans.**	
Workpackage	WP7	Living Lab Automotive
Task / Use Case	T7.1-T7.6	All tasks in WP7
Subtasks involved	All subtasks in all tasks in WP7	
Lead contractor	Infineon Technologies AG Werner Weber, werner.weber@infineon.com	
Deliverable responsible	VOLVO Thomas Söderqvist, thomas.soderqvist@volvo.com	
Version number	v1.0	
Date	2015-04-02	
Status	Final	
Dissemination level	Restricted (RE)	

Authors

Partici-pant no.	Part. short name	Author name	Chapter(s)
16F	VOLVO	Thomas Söderqvist	1, 2.1, 3, 2.7, Appendix 6
17I	Technolution	Dave Marples	2.2, Appendix 1
15M	IXION	Jorge Villagra	2.3, Appendix 2
02A	AVL	Ismar Mustedanagic, Eric Armengaud	2.4, Appendix 3
11B	CRF	Alberto Melzi	2.5, Appendix 4
13B	CSOFT	Jorge Almeida	2.6, Appendix 5
16F	VOLVO	Mafijul Islam	2.7, Appendix 6

Document History

Version	Date	Author name	Reason
1	2015-03-31	Thomas Söderqvist Dave Marples Jorge Villagra Eric Armengaud Alberto Melzi João Pedro Rodrigues Mafijul Islam	
v1.0	2015-04-02	Alfred Hoess	Final editing and formatting, deliverable submission

Table of contents

List of figures

List of tables

1. Introduction

1.1 Objective and scope of the document

This document gives a high level overview of all six automotive use cases and their subtasks. For each use case the business needs, KPIs, high level requirements and evaluation plans are described. The objective of this document is to set the frame for WP7 and for each use case. The content of the use case design deliverables D7.3, D7.5, D7.7, D7.9, D7.11 and D7.13, one per use case, shall be seen as refinements and use case specific solutions to the business needs and high level requirements expressed here in D7.1.

In addition the high level requirements shown here are handed over to the technology work packages WP1 – WP6 for analysis, refinement and for creation of general solutions that can be adapted to use cases in WP7 – WP11.

The following use cases and lead partners are covered:

Table 1: Automotive use cases and partners

Use case	Lead partner	Participants
ADAS and C2X	TECHNO	BMW, DENSO, EB, NXP-GE, IFAT, TTT, HIB, NXP-NL, TUE, TNO, TOMTOM
Highly automated driving	IXION	HUA, HIB, UoMAN, CHALMERS
Design and validation of next generation hybrid powertrain / E-Drive	AVL	IESE, IFAT, AIT, IFAG, EB, SFR, TUW, TNO, BUT, Tecnalia, VIF
Modelling and functional safety analysis of an architecture for ACC system	CRF	DITEN
Infotainment and eCall Application Multi-Critical Application	CSOFT	AICAS, CISTER, HIB, INESC-ID
Next generation electronic architecture for commercial vehicles	VOLVO	KTH, Chalmers, ArcCore, ViF, TTTech, SICS, Magillem, Systemite, ALTEN, IFAT

1.2 Structure of the deliverable report

The main body of this document contains a common introduction for the automotive Living Lab. Next part contains for each use case an abstract and the driving business needs.

More details, like use case subtask descriptions, high level requirements, KPIs and evaluation plans for the high level requirements are described in one appendix per use case.

The intention with this structure is to keep the main body of this document limited in size and the reader can get a more complete overview of all use cases by reading just this main part. Further details about the use cases are given by reading the appendices.

2. WP7 Automotive Use Cases

2.1 Introduction

The driving sources for the business and technical needs in WP7 are the automotive use cases within different technical areas. Common technical topics within the automotive Living Lab are:

- ADAS, Advanced Driving Assistant Systems, Highly Automated Driving
- Next generation of hybrid powertrains
- Infotainment and SW defined radio
- Next generation embedded electronic architecture for commercial vehicles
- Optimizations on AMSPS and Power Management
- Run time optimization of fast control algorithms
- EV and HEV energy recuperation

The automotive living lab use cases will leverage the technological advances developed in the WPs and particularly contribute to showcasing innovations in the following areas:

- Novel ADAS following large-scale software integration and full exploitation of available resources.
- Development, deployment and practical evaluation of Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) applications.
- EMC2 Electronics Power Management: Requirement collection, concept development and HW demonstration.
- Master the next challenges on the road to fully automated driving by optimizing the use of HW/SW resources of the proposed multi-core embedded cloud and exploiting them in a real-life autonomous driving scenario.
- Next-generation hybrid powertrain control that are built following model-based engineering and new programming paradigms, and are supported by simulation.
- Develop and showcase new architectural solutions for electric and hybrid vehicles, in particular researching and developing contactless charging solutions.
- Demonstrator development realizing and evaluating an integration of mixed criticality applications using mainly COTS.Harvest the potential of multicore CPUs for mixed-critical applications in the next generation ofElectrical and/or Electronic (E/E) architecture for commercial vehicles.

For an extensive set of technologies related to WP7 and the automotive partners please read the exploitation plan described in the EMC2 deliverable D13.16 – Exploitation Plan V1.

2.2 T7.1 UC ADAS and C2X

2.2.1 Abstract

Cars and roads are becoming smarter. Communication technologies are emerging that connect the cars and the road-infrastructure enabling so-called cooperative applications. In parallel, Automotive technologies are emerging that enable cars to take over driver tasks more and more with the ultimate goal to create fully autonomous driving cars. Further, while this migration continues, there remains considerable pressure to reduce the cost and improve the energy efficiency of in-vehicle systems.

It is within this context that the ADAS and C2X use case has been constructed. The objective of this use case is to investigate and implement a system architecture that combines the potential of existing and emerging technologies for markets like ADAS and next generation vehicle infrastructure. Our goal is to deliver strategies for addressing these markets with new communication and MCMC architectures using tested and field qualified approaches. Simultaneously we must recognise and address the cost pressures associated with the in-vehicle systems supporting these applications. EMC2 architectures offer an approach to addressing these cost pressures by combining multiple functionalities onto a single platform, but this needs to be done in the context of the mixed criticality constraints that inevitably arise in consequence.

This objective will be met via hands-on experience in real-life trials; both the in-vehicle and ADAS systems demand critical software components but cannot carry the cost (both in terms of time to market and financial) of a complete safety-critical structure for all aspects of the environment. They therefore will demand a mixed-criticality infrastructure with critical and non-critical functionalities partitioned in a provably correct manner.

The demonstrator comprises two distinct groupings by virtue of the partners that are involved and their interest focus, pulled together by their common recognition that ADAS represents the future. One of these groupings is concentrating on the in-vehicle systems and the migration they have to perform, and centred on EMS and EV Charging, the other one is concentrating on the ADAS and Collaborative Systems aspects. The use case has delivered requirements to WP1 and WP3 and is expected to exploit results from these technical work packages.

Figure 1 below shows the relationship between the ADAS systems and the vehicle hardware. The Vehicle State Estimator is the component that holds these two elements together.

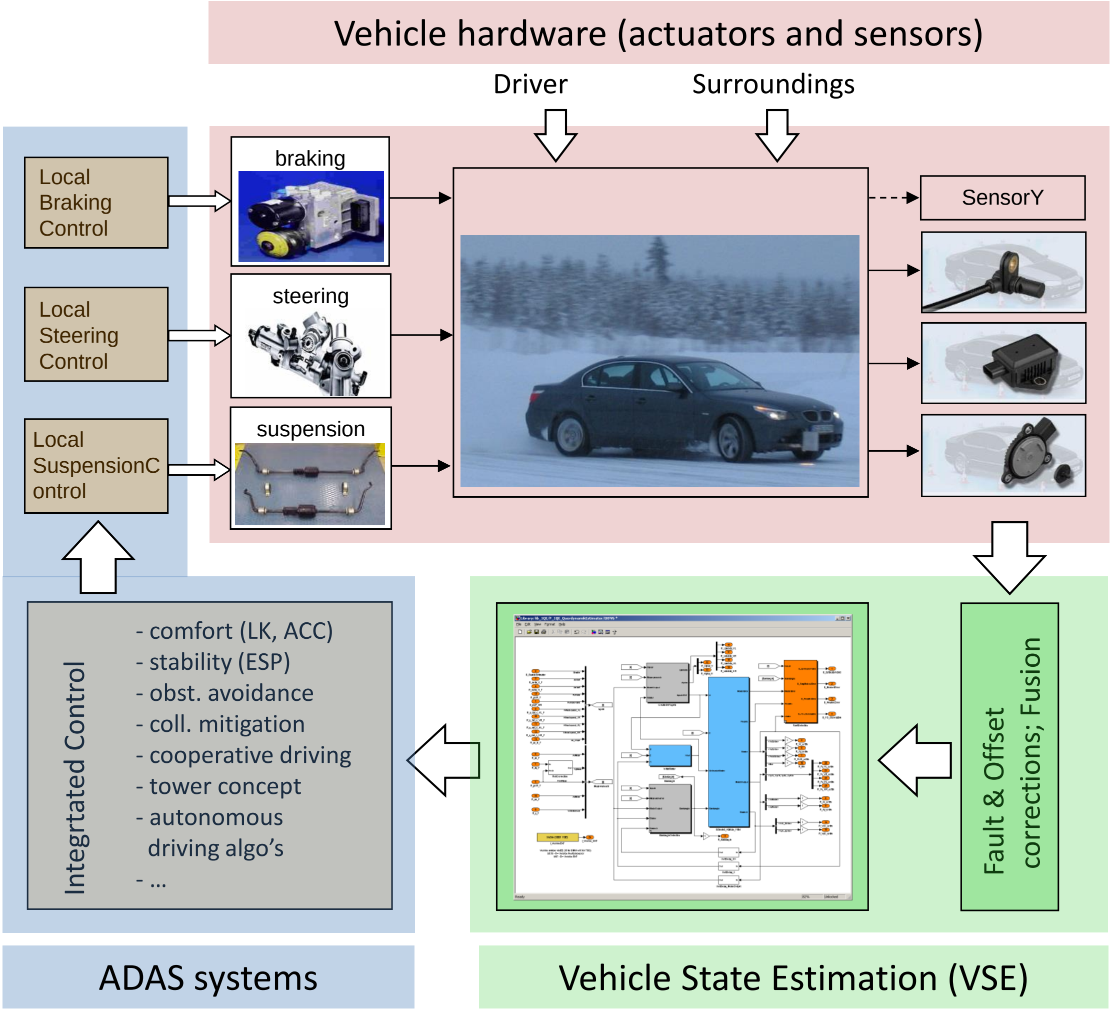


Figure 1: Relation of the subsystems in an ADAS/C2X environment

The use case will deliver:

- Novel ADAS following large-scale software integration and full exploitation of available resources
- Development, deployment and practical evaluation of Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) applications, supported by back end infrastructure.
- An understanding of how multicore systems can be applied to achieve cost reductions and/or reliability improvements for traditional EMS applications.
- An understanding of how mixed criticality systems can be deployed into the 'traditional' in-vehicle environment in the light of the increased demands that arise from ADAS and the migration to the connected vehicle.

The Use Case explicitly leverages the results from the technical work packages, especially WP3, for the purpose of creating the technical platforms deployed in the vehicle and at the roadside.

2.2.2 Business needs

Table 2: Business needs for use case "ADAS and C2X"

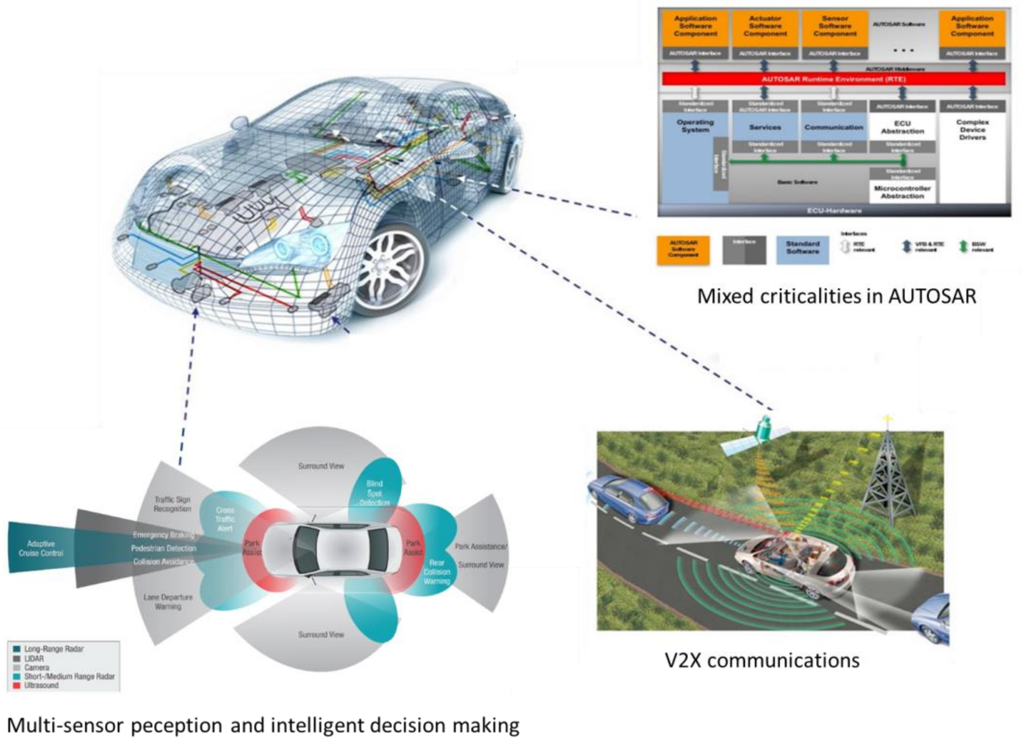
No	Title	Description
BN_T7.1_01	Testing and Validation	Testing and validating ADAS systems is getting more complex due to their increased dependency on environmental conditions that cannot be exhaustively tested and their interaction with other ADAS systems that control on the same actuators (brakes, throttle and steer). The test costs should be kept limited while at the same time they should give more confidence in the ADAS performance (false positives/negatives) under all possible conditions.
BN_T7.1_02	Product Offer	The product offer for an ADAS system must be compelling in order to be able to sell them in a competitive market environment. They should be able to maintain and extend them in a cost-efficient manner.
BN_T7.1_03	Real Time Response	Increasingly complex software infrastructures demand realtime responses which cannot be adequately addressed by operating systems running on time-sliced single CPUs – especially when new demands a generated from outside the vehicle (i.e. from ADAS connectivity).
BN_T7.1_04	Mixed Criticality Software Stacks	Different types of software stacks demand different development methodologies with widely different cost implications. This is clearly illustrated by safety critical and non-safety critical code. Means need to be found to reduce the overall cost burden of code development while not compromising the performance or safety of the overall system.
BN_T7.1_05	Safety, Privacy and Security	ADAS systems must not compromise safety or security and should maintain users privacy to the maximum extent permissible by law – this is particularly relevant when the interaction between ADAS and the existing on-vehicle systems is considered.
BN_T7.1_06	Legacy Migration to Multi-core	The use of multi-core systems for high energy efficiency and performance requires the parallelisation of existing legacy applications to save the cost of new development and improve time to market. These applications still have to accommodate the new demands for ADAS functionality.
BN_T7.1_07	Dynamic and Safe RTE	It should be possible to integrate new software needs into these in-vehicle systems during their lifetime. The testing and qualification process is far more complex when ADAS functionality is involved, especially when the controller can contain code from different suppliers.

The exploitation of the results from this Use Case vary by partner, but inevitably center around the technical migration from multiple seperate ECUs to multi-function, multi-CPU modules. Some partners will exploit the results through a better understanding of how to migrate existing codebases to this new environment. Some will exploit them through the creation of new silicon and market approaches and some view the embryonic market for ADAS as a direct growth opportunity.

2.3 T7.2 UC Highly automated driving

2.3.1 Abstract

The introduction of telematics in the latest generation of advanced driver assistance systems is one of the most challenging differentiating factors that would transform cars into intelligent personalized comfortable spaces. The evolution towards integrated and dependable ADAS systems will accelerate the trend towards zero-accident vehicles, highly automated driving and an associated road-utilization. Preventing accidents and automating tasks in progressively more complex driving situations requires innovation in perceiving the environment, the driver-state, the vehicle-state, and the reasoning/interacting about these tasks. This will be accomplished by the usage of multiple sensor technologies such as computer vision systems and lidars, complementing and reinforcing each other. The information these physical sensors provide will be complemented and reinforced by information communicated by and to vehicles using cellular (3G, 4G LTE) and dedicated short range communication technology (IEEE 802.11p), as can be seen in Figure 2.



Mixed criticalities in AUTOSAR

V2X communications

Multi-sensor peception and intelligent decision making

Figure 2: Highly Automated Driving Use Case concept

In addition, since the human intervention might still be necessary in specific situations, advanced human machine interfaces have to be considered not only for safety-related feature, but also for other purposes such as navigation or web services. Indeed, an important number of advanced services based on vehicles connectivity are emerging. Dynamic ridesharing and intelligent parking management are two significant examples of the upcoming associated value added that highly automated vehicles can offer to their users.

A commercial vehicle includes nowadays time-critical embedded control-units (engine control, ADAS,…), and a platform for entertainment (navigation, multimedia or even cloud-based dynamic services). The state of the art automotive embedded systems are based on dedicated ECU for every purpose, each of which is subject to a series of functional and safety requirements. As a result, today's modern vehicles can carry up to 100 ECUs on board (e.g. BMW 7). This creates a several disadvantages, namely material costs, communication bandwidth limitations, latency inconveniences, higher power consumption. Derived from the high complexity and logical and physical interfacing requirements, also potential robustness issues and high developments and maintenance costs can occur.

In a parallel line, the progress in standalone and cooperative control systems towards highly automated vehicles is becoming a reality in experimental prototypes. It is then mandatory to propose adapted solutions that take into consideration the heavy computational payload of such systems, adaptively combine it with the existing platforms and significantly reduce the number of ECUs.

The proposed novel architecture of EMC2 aims to find a solution to the observation that with the increasing number of ECUs the current approach is hardly sustainable. Additionally, a key issue for driver acceptance and quick market introduction of such real time complex controlled systems is to assess its performance under uncertainty. As a result of this, new functional safety and integrity requirements are appearing (ISO 26262). In consequence, the main objective of this Use Case is to use EMC² architectures and tools in real-life tests to develop a highly integrated computing unit, using current advanced sensing, navigation, decision making and cooperative functionalities for highly automated vehicles. The resultant design will include a SW architecture enabling the implementation of efficient algorithms in multi-core environments while preserving a high level of safety and reliability

More specifically, it will be required to integrate multiple existing systems with different characteristics in terms of computational cost, control cycle period, or level of criticality (ISO 26262, IEC 61508). As a result, (i) an increased integration together with a (ii) consistent separation will be natural requirements for the resulting mixed criticality runtime environments.

(i) **Increased integration**: Highly intensive computational components such as perception, map-based localization, decision making and V2X communications (see Fig. 2) are proposed to be integrated in a reduced set of computing platforms or in a highly integrated ECU, using a hybrid combination of different technologies witin the same HW platform. The existing set of verification & validation artifacts for isolated components will be exploited at best for the resulting configuration in order to reduce V&V costs and efforts.

(ii) **Separation**: One of the objectives for a mixed-criticality solution is to provide sufficient separation between elements of differing criticality levels to guarantee that the lower criticality elements cannot interfere with the functioning of the higher criticality elements. In particular, navigation related SW components will have a lower criticality than most of the embedded subsystems. Fig. 2 shows that this separation can be done considering functional aspects in most of the cases. However, there are some specific components that combine high and low critical systems (namely, the decision system, which includes a time-critical motion planner, and a learning process that can be handled with a much lower criticality). System fault tolerance and resources requirements at local level will be taken into consideration with modelling approaches.

The main objective of this use case will be to investigate, implement and evaluate a system architecture to exploit at best the potential of existing technologies around highly automated driving. To that end, the EMC2 architecture and tools should enable scheduling time-critical and less time-critical high-performance functionalities on the same system, which will be tested in real driving scenarios. To that end, an urban commuting scenario has been identified, where safety for driver/passengers and vehicle behaviour predictability will have to be guaranteed both for drivers and for other road users to let them trust in these automated systems. Both simulation and real-life experiments will be made to show the potential of new multi-core service oriented architectures in the context of highly automated and connected driving.

The functional architecture depicted in Figure 3 permits to understand the specific areas of interest for this Use Case:

- Implement both existing and enhanced embedded functionalities as AUTOSAR SW compliant components.
- Integrate and validate V2I-based navigation related functionalities (Dynamic ridesharing, Intelligent Parking Management) and V2X safety components, putting special focus on mixed criticalities handling.
- Explore the advantage of modelling/simulation tools from different perspectives:
  - Modeling behaviors of hierarchical, compositional, and executable components within SOA architecture in a multi-core target.
  - Setting up a simulation framework that permits to validate specified functionalities in the designed embedded system (under a HIL scheme) within a realistic traffic simulation environment, where web-based services as Ridesharing can be properly integrated.
- Investigate the possibilities of EMC2 architecture and tools for highly automated vehicles on heterogeneous platforms (combining multi-core CPUs with GPUs or FPGAs)
- Properly model mixed criticalities both of the whole chain value and within individual software components (e.g. Decision System in Figure 3 includes a critical motion planning subsystem, and a less critical one in charge of on-line automatic learning mechanisms)
- Analyze multi-sensor perception limits (number and nature of sensors) for a specific target and real-time constraints.
- Optimize the available resources to guarantee the critical components to be safely scheduled within the runtime environment in any operation mode (including graceful degradation modes)

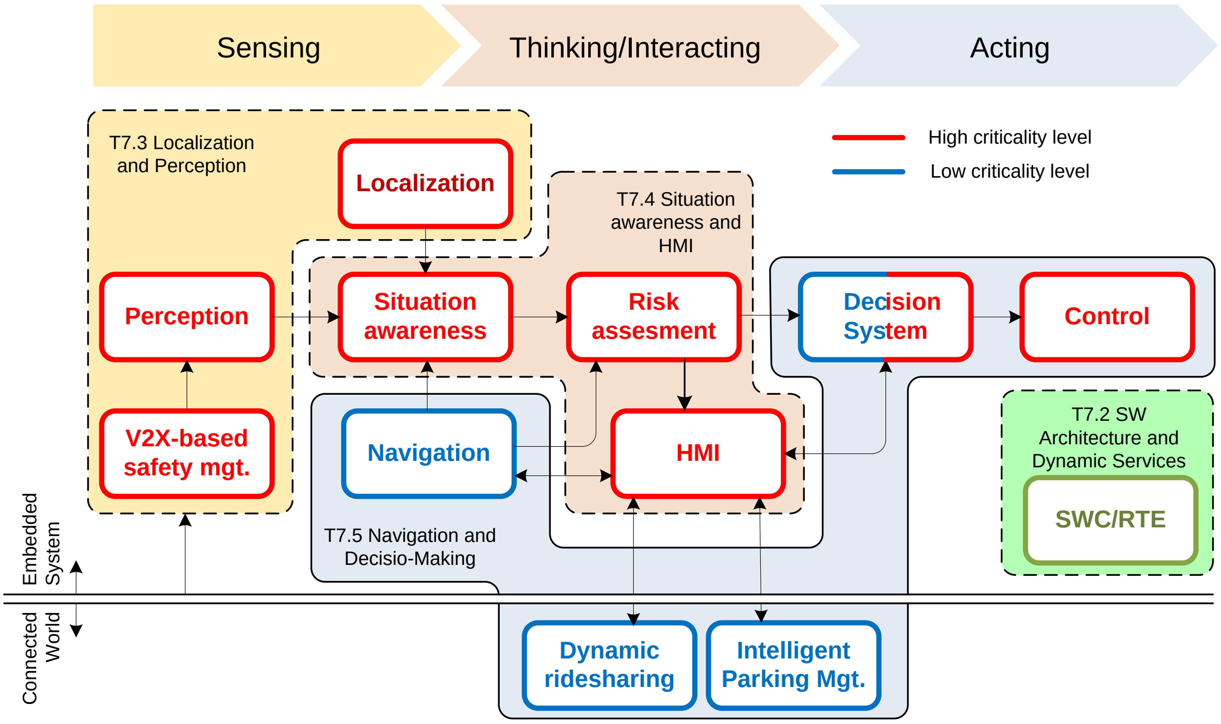


Figure 3: Functional architecture of Use Case Highly Automated Driving

2.3.2 Business needs

Table 3: Business needs for use case “Highly automated driving”

No	Title	Description
BN_T7.2_01	Modeling and simulation environment for multicores	Enhancement of simulation environments to properly evaluate High Level functionalities within the adopted multicore SOA architecture, integrating performance and power modeling capabilities.
BN_T7.2_02	Harmonization of development tools	Availability of a set of harmonized tools that allow to easily integrate the design and development work flow, from system to component/code level.
BN_T7.2_03	Effective HW/SW Integration in heterogeneous platforms	Exploration of methods and tools able to increase confidence (reduce risk) of problems being observed at integration time, taking into consideration all SW layers.
BN_T7.2_04	Resource Optimization	Dynamic optimization of computational resources, hence energy consumption and costs, according to the performance requirements without putting quality at risk.
BN_T7.2_05	Effective Portability	Characterization of software components on one hardware platform so that they can be located on different hardware platforms without running the full set of timing and analysis on each platform; this supports a higher flexibility in the design of the system.
BN_T7.2_06	Parallelization and classification	Development of functionalities able to (i) recognize computing structures of application source code that may be suitable for parallel execution and (ii) to classify and evaluate different parts of the algorithms and decide if they can be effectively moved to any accelerator component of the current HW.

2.4 T7.3 Design and validation of next generation hybrid powertrain / E-Drive

2.4.1 Abstract

Desktop computers are already integrated with multi-core processors for several years due to continuously increasing performance requirements and stricter power limitiations. This trend is reflected in modern times also in the automotive field and for chip vendors by releasing automotive embedded system architectures based on multi-core technologies to counteract on increasing performance demands. Another major aspect for this upcoming trend is the increasing amount of ECU’s within the vehicle; Modern vehicles are equipped with 70 to 100 ECU’s communicating trough the existing networks within the vehicle with each other to handle the necessary control SW system for vehicle operation. With upcoming multi-core technologies this amount could be reduced by combining different control application with possibly mixed criticality into one multi-core ECU.

The major objectives within this use case are the design and validation of several system architectures based on multi-core platforms in order to be able to handle this future trend in the automotive field. The use case addresses the following main topics in an existing hybrid/electrical powertrain application:

- Integration of multi-core technology in existing control applications in order to provide more computing resources for (a) improvement of existing functions and (b) new functionalities.
- Combining two automotive controller units in one (EMCU + VCU = VEMCU): Electrical and functional integration of high dynamic e-drive system controls with time based vehicle control algorithm.

To achieve these objectives several technical aspects need to be analyzed for these new multi-core technologies:

- Integration of mixed criticality SW components on multi-cores

- Integration and handling of AUTOSAR for multi-core architectures
- Safety aspects for multi-cores
    - Program flow of software components executing on different cores
    - Data sharing strategies between OS tasks and cores
- Simulation concerns for applications running in parallel
- New networking solutions for the increased amount of ECU data

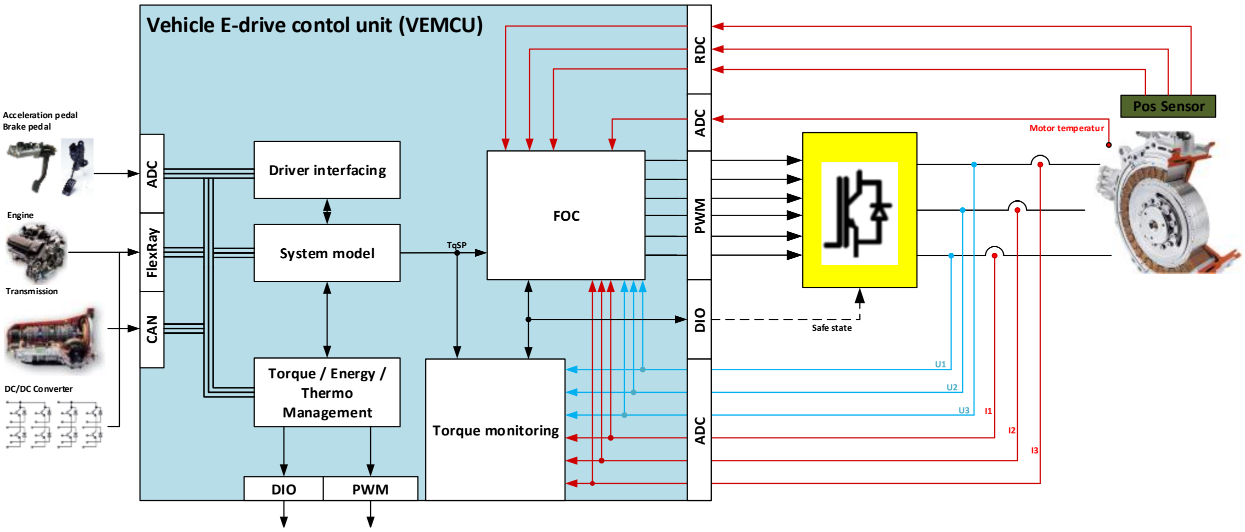


Figure 4: Overview of Vehicle Electric Motor Control Unit

Figure 4 provides an overview of the vechicle e-drive control unit (integrating the vehicle control unit VCU and the e-drive control unit ECU). More especially, the different actuators are listed on the left part of the figure and consist on the accelerator and brake pedals, the combustion engine sensors, the transmission status (over CAN) and the DC/DC converter. Further, the interface signals to the e-drive are listed on the right part of the figure. Finally, the main control strategy architecture is depicted in the center of the figure.

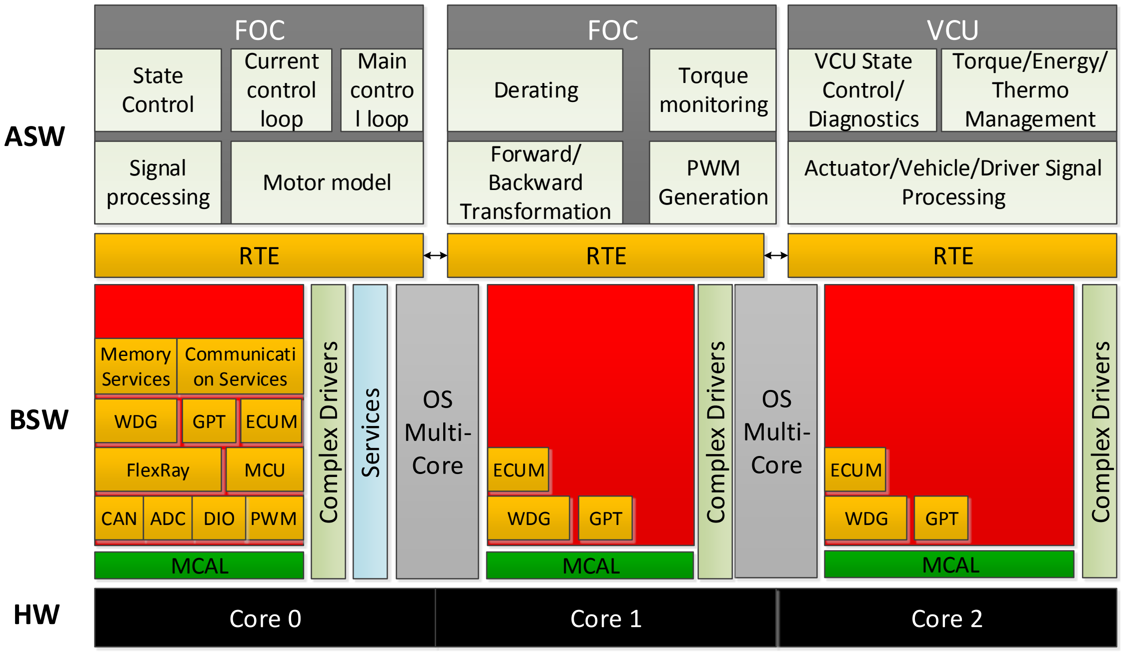


Figure 5: Overview of software distribution and interaction between cores

For this use case, a AURIXTM automotive CPU from Infineon is planned. Figure 5 describes the planned SW allocation to the different CPU cores. More especially, main control strategy for the e-Drive is allocated to Core 0, main control strategy for the vehicle is allocated to Core 2, and the I/O coordination and resource management is allocated to Core 1.

2.4.2 **Business needs**

The following table identifies all business needs which shall be addressed by this use case. Each sub task in the use case shall be planned and executed in such a manner to fulfill individual business needs to a certain extent.

Table 4: Business needs for use case "Design and validation of next generation hybrid powertrain / E-Drive"

No	Title	Description
BN_T7.3_01	Simulation environment for multicore application	Enhancement of simulation environments to emulate parallel applications for early architecture exploration and improved safety analysis
BN_T7.3_02	Seamless modeling method	Capability for seamless architecture modeling including system, SW, HW, behavior and safety aspects
BN_T7.3_03	SW design and development guidelines for highly parallel HW platforms	Availability of guidelines for SW architecture / SW development including all (AUTOSAR) relevant SW layers
BN_T7.3_04	Tailored safety architectures for multicore platforms	Availability of new safety concepts, architectures & services for multi-core platforms
BN_T7.3_05	Integrated tool chain	Availability of integrated and consistent tool chains for control system development
BN_T7.3_06	Safety case generation	Automated compilation of safety case information based on existing information from product development
BN_T7.3_07	New networking solutions for multicore platforms	Availability of improved networking solutions for heterogeneous automotive systems
BN_T7.3_08	Data exchange and communication strategies	Availability of new solutions for the configuration and handling of complex communication channels during development, calibration and diagnosis

2.5 T7.4 Modelling and functional safety analysis of an architecture for an ACC system

2.5.1 **Abstract**

Novel architectural solutions for advanced vehicles shall be designed to be compliant with ISO 26262 standard, therefore these solutions shall be modelled and analysed in conformity to ISO 26262 requirements. Objective of this use case is to run the analysis and the modelling of architecture for ACC from a functional safety perspective, considering the implication of the ISO 26262 standard and the related verificationat concept level, and to evaluate the modelling methodology applied using a semi-formal language. This aim implies the structuring of the ISO 26262 standard requirements involved in the task, aiming to reach some degree of automation into the applied process. The expected outcome is the evaluation of the modelling methodology at the concept (functional/technical) level, based on the ACC architectural modelling, according to ISO 26262 standard process. This methodological outcome will enable an improved design for future ISO 26262 product oriented solutions, with the aim of increasing confidence/robustness and reducing cost/time to market of the functional safety process.

2.5.2 Business needs

Table 5: Business needs for use case "Modelling and functional safety analysis of an architecture for ACC system"

No	Title	Description
BN_T7.4_01	Modeling environment for ISO 26262 functional safety analysis of safety mixed critical systems.	Improvement of modeling/simulation environment to integrate safety mixed critical system architectures and requirements within ISO 26262 functional safety analysis constraints.
BN_T7.4_02	ISO 26262 compliant safety case generation.	Automated compilation of the safety case structure based on the results from the above defined modeling/simulation environment, according to ISO 26262 standard.
BN_T7.4_03	ISO 26262 functional safety assessment automated / semi-automated.	Automated/semi-automated ISO 26262 standard assessment of the functional safety analysis from the modeling/simulation environment and safety case above defined.

2.6 T7.5 Use case Infotainment and eCall Multi-Critical Application

2.6.1 Abstract

Several functionalities require distinct criticality levels in the embedded world. Domains like automotive, aerospace, space, medical and defence are used to have different criticality levels for each selected functionality that greatly influence their development and certification costs.

On of the most important aspects of critical systems is certification. Most applications developed with safety critical aspects must convey to some set of norms and/or standards. To raise the levels of safety assurance of a specific feature requires the development and validation process costs to rise as well.

In a non mixed-criticality system the certification process usually requires that all the applications conform to the same level of criticality on the platform, creating the need for system isolation. This requires a hardware partitioning on systems offering functionalities of distinct levels of safety-critical requirements, such that no interference occurs between the different criticality levels. Using the mixed-criticality system's approach, one must first certificate the system, a combination of hardware and software base platform and the supporting functions. Subsequent applications shall inherit the base platform certification and will only be analysed/certified the ones that are safety critical, according with the applicable Automotive Safety Integrity level.

Critical and non-critical functionalities typically require different set of skills and knowledge to be implemented. The application software development of mixed criticality applications can be made more efficient by choosing a suitable platform, for example, using a spatial and temporal partitioned kernel like the ones provided in LynxOS, INTEGRITY or VxWorks. Having a platform that supports both criticalities with little or no effort is crucial for the development of mixed criticality applications in the same platform, being the concept of platform a wide term (using for example distributed processing capabilities). As an example, take for instance the application's Graphic User Interface (GUI). The application's GUI have two very different paradigms in the critical RTOS and non-critical feature OS. Critical RTOS interfaces tend to be simplistic and low-level due to the need for determinism, on the other hand, in the feature OS the interfaces tend to be feature rich and extensive with a greater selection of supporting libraries, but on a best effort policy. This differences cause a greater level of effort to develop the same GUI in the RTOS than on the featured OS.

There are a number of other possibilities to increase efficiency of application development. By using higher level programming languages and access to libraries and frameworks not available in the critical domain mostly due to their certification costs. The use of these features allow for a faster development and integration by using commercial off-the-shelf (COTS) components that handle common tasks for the developers such as, window composition, image and video processing, database engines, among others are not present in most safety critical RTOS.

Another potential gain by using mixed-criticality systems, is the reduction of overall number of components and by so, reduce failures (ISO26262 objective) and ultimately, weight. According to the ACEA1 statistics, the average European vehicle travels 14000km every year and has an age of 8 years. Modern vehicles have in excess of 50 microcontrollers for the electronics subsystems, these only weight a few grams each but the necessary amount of fuel and braking power to accommodate these few grams are quite big when taken in account the entire lifecycle of the vehicle. Associated with these systems there is also a good deal of electronic systems repeated like: housing, power supplies, voltage regulators, buses and others that can be reduced by merging several of them in the same physical platform, or reusing the same HW to do similar tasks by different applications, with the support to have a mix-criticality system. Taking this into account along with the power needs of such devices, the points of failure introduced by multiple devices and multiple buses, it is possible to reduce the number and weight of batteries, the amount of cabling and therefore the weight of the vehicle making it more efficient and less costly.

Hypothetically, let’s consider a specific vehicle, weighing 2000 Kg, which uses 75 microprocessors and 40 different cables sections to connect the microprocessors to sensors, actuators and inter-system communication. By using a multi-criticality system we reduce the number of microprocessors and cable section to save 5 Kg.Assuming a 0.055 L/100Km average fuel consumption for each kilogram, we would get an average fuel consumption of 0.0548625L/Km/Kg. This reduction is the equivalent of an increase in efficiency of 0.25%. Over the lifetime of the vehicle (8 years) we could observe a savings in fuel usage of 15.4L.

Another important aspect is the reduction of the number of ECUs in the vehicle that indirectly will provide a total reduction of production pollution and reducing the total cost of producing the vehicle. With less ECUs in the vehicle will also diminish the need for electric poer and consequently the usage of larger batteries.

Majority of the automotive systems are composed of several systems of mixed criticality by nature. One cannot compare the level of criticality of functionalities of the Brake Control Modules with the Navigation System, even though they can share hardware resources. Like these two systems we could name countless other examples like the instrumentation cluster, the engine control modules and the audio system, etc. We propose an architecture for a service oriented architecture taking into account the results of the research project (xLuna). This architecture will enable the hosting of a feature rich OS, with the full plethora of libraries and frameworks, running concurrently with a safety critical RTOS. The RTOS will handle all the accesses to the critical and/or shared resources and will execute the critical functionalities, in the local HW or in remote HWs according with the developed SoA architecture for the EMC² project.

The distributed and multi-criticality architecture allows for the efficient partitioning of the non-critical functionality and the Real Time and Safety guarantees by the RTOS, at the cost of performance of the non-critical sections. This approach as several advantages:

- Only the critical functionality will be developed using Real Time constraints;
- The non-critical functionality will have access to the plethora of libraries and frameworks available for rapid development;
- The non-critical Feature OS can be managed by the RTOS and by so guaranteeing the safety requirements;
- Improved load handling on multi-core systems, not relying on core separation or local HW for load execution.

This UC will be focused on the aspects related with the multi-criticality of an infotainment system, a non-critical application, and a safety critical application based on an eCall platform. Since these two

[1] http://www.acea.be/news/news_detail/vehicles_in_use/

applications share many HW resources, like GPS, GSM connection among others, this use case is expected to demonstrate the full capabilities of the EMC² framework in a real problem in the automotive domain.

The system will also provide for dynamic extension and update. This will be done based on an application framework which can either be in the guest OS for non-critical applications or in the base OS for critical applications. The app framework will support fine-grained resource access and isolation between apps, thereby providing higher security than current app platforms such as Android or iPhone.

The results of this use case are expected to be reused by the industry in the near future with the objective of reducing the number of physical ECUs inside a vehicle, and consequently reduce the fuel consumption and the cost of vehicle construction.

2.6.2 Business needs

Table 6: Business needs for use case "Infotainment and eCall Multi-Critical Application"

No	Title	Description
BN_T7.5_01	Mixed criticality	To reduce certification costs, the system must support execution environments of different levels of safety integrity levels.
BN_T7.5_02	Multicore CPU	To address mixed criticality environments and the increasing concurrent tasks paradigm infotainment systems require, the system must allow concurrent mixed-criticality software execution.
BN_T7.5_03	Secure communication	Both critical and non-critical environments must be able to share data.
BN_T7.5_04	General purpose OSs	In order to allow an easier and faster development platform a general purpose OS shall be used for the infotainment GUI.
BN_T7.5_05	Real Time OS	Critical tasks require hard real time scheduling and shall be managed by a RTOS.
BN_T7.5_06	Infotainment	The system shall provide a graphical user interface with infotainment capabilities, e.g., multimedia, GPS navigation and internet connectivity.
BN_T7.5_07	System peripherals	The system should support touchscreen, 2D/3D HW accelerated GPU, 3 axis accelerometer, GPS, sound card and a GSM/3G modem.
BN_T7.5_08	eCall system	The system should provide the functionality to implement an eCall emergency system.
BN_T7.5_09	Hardware Assisted Virtualization	To improve performance and reduce development and certification effort the hardware shall support virtualization extensions, such as ARM's TrustZone.
BN_T7.5_10	Resource sharing	The system must allow for individual HW resources to be partitioned and/or shared between environments of distinct criticality
BN_T7.5_11	Resource criticality reassignment	The system must allow for a HW resource to be repurposed from a lower criticality environment to a higher on in case of an event occurred.

2.7 T7.6 Next Generation Electronic Architecture for Commercial Vehicles

2.7.1 Abstract

The goal of this use case is to identify, apply and evaluate methods as well as tools to harvest the potential of multicores for mixed-critical applications in the automotive industry. Main topics of interests are to explore the full potential of multicores based on the needs and requirements of the next generation Electrical and Electronic (E/E) architecture for commercial vehicles, for example, trucks and buses. In particular, this task will focus on:

- Efficient E/E architecture, including communication aspects (logical and physical architectures, mapping between logical and physical architectures, qualitative and quantitative evaluation of alternative architectures) based on multicores.
- Efficient partitioning and allocation of functionalities as well as software on multicores, including separation among elements of different criticality levels and management of shared resources.
- Improvement of dependability (functional safety, reliability, fault tolerance, robustness, availability, security) by using multicores along with support for scalability, flexibility and adaptability.
- Applicability of Service oriented Architecture (SoA) and run-time optimizations for real-time safety-related systems based on AUTOSAR [1].

Figure 6 graphically represents the objective of the use case: how to achieve the overall goals for a given set of requirements and constraints, assuming two abstraction levels:

A. E/E system architecture consisting of a set of multicore ECUs in addition to other components, and
B. A single multicore ECU within the E/E system.

Requirements consists of both functional (e.g., performance) and non-functional (e.g., functional safety) requirements whereas constraints may encompass cost, legacy, technological trends, available hardware resources (e.g., computation power, memory, communication bandwidth) and market trends (e.g., unanticipated new features). The objective is then to identify an E/E architecture that (a) fulfills the functional and non-functional requirements, (b) supports scalability, flexibility, adaptability and reusability, and (c) finds the best possible alternative by applying multi-objective optimization, considering the constraints. Note that this use case consists of several main topics of interests towards the common goal of exploiting the potential of multicores. To achieve the objective and make progress beyond state-of-the-art, the following innovation activities are planned to be performed within the T7.6:

- Define, develop and apply metrics as well as methodologies for systematic evaluation of electronic architecture alternatives. Embedded vehicle architecture concepts based on computation nodes and generic I/O nodes
- Efficient methods and tool support for partitioning, allocation, mapping and scheduling of software on multicore ECUs.
- Develop tools and techniques to exploit multicore systems properly taking into consideration resource sharing issues among cores. Methodologies for enforcing strict isolation in a multi-core system.
- Service oriented Architecture (SoA), dynamic configuration of services and their optimization during runtime, considering real-time requirements.
- Methodologies for analyzing multi-core ECU configuration with respect to compliance with safety/dependability requirements. New algorithms for efficient detection of faults on multi-core platform and new methodology of fail-safe or fail-operational system development.

Relevant standards and specifications such as ISO 26262 and AUTOSAR are central to the use case with respect to not only considering present requirements on the E/E architecture and multicores but also adapting these standards to state-of-the-art in multicore systems with mixed-criticalities.

The EMC2 deliverable **D7.13 – Preliminary UC T7.6 design** presents more detailed information about the preliminary use case design of the T7.6. The exploitation plan for the T7.6 has been described in the EMC2 deliverable **D13.16 – Exploitation Plan V1.**

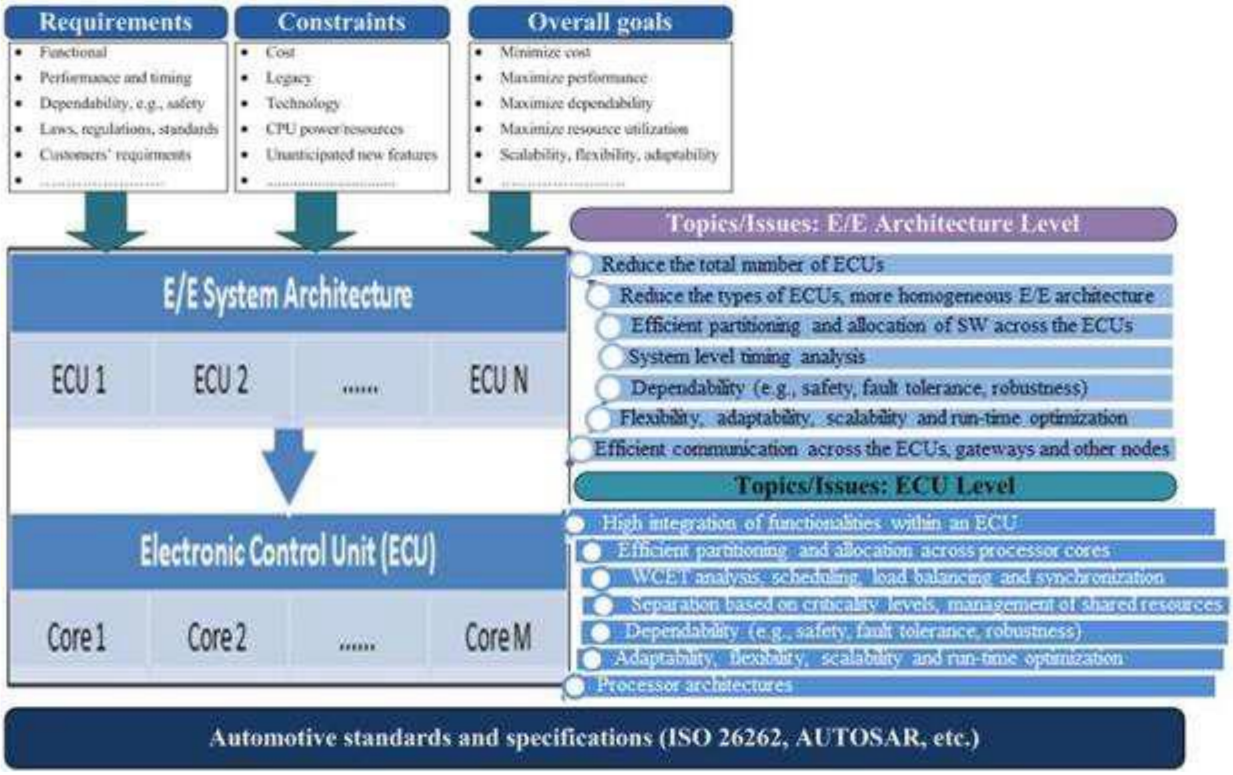

Requirements
Functional
Performance and timing
Dependability, e.g., safety
Laws, regulations, standards
Customers' requirments
Constraints
Cost
Legacy
Technology
CPU power/resources
Unanticipated new features
Overall goals
Minimize cost
Maximize performance
Maximize dependability
Maximize resource utilization
Scalability, flexibility, adaptability
E/E System Architecture
ECU 1
ECU 2
......
ECU N
Electronic Control Unit (ECU)
Core 1
Core 2
......
Core M
Topics/Issues: E/E Architecture Level
Reduce the total mmber of ECUs
Reduce the types of ECUs, more homogeneous E/E architecture
Efficient partitioning and allocation of SW across the ECUs
System level timing analysis
Dependability (e.g., safety, fault tolerance, robustness)
Flexibility, adaptability, scalability and run-time optimization
Efficient communication across the ECUs, gateways and other nodes
Topics/Issues: ECU Level
High integration of functionalities within an ECU
Efficient partitioning and allocation across processor cores
WCET analysis, scheduling, load balancing and synchronization
Separation based on criticality levels, management of shared resources
Dependability (e.g., safety, fault tolerance, robustness)
Adaptability, flexibility, scalability and run-time optimization
Processor architectures
Automotive standards and specifications (ISO 26262, AUTOSAR, etc.)

Figure 6: Overall objectives of use case "Next Generation Electronic Architecture for Commercial Vehicles"

2.7.2 Business needs

The use case will address the following issues (business needs) in relation to the next-generation E/E architecture for commercial vehicles and a subset of the ECUs specific to the engine controllers:

Table 7: Business needs for use case "Next generation electronic architecture for commercial vehicles"

No	Title	Description
BN_T7.6_01	Number of control units	Reduction of the total number of ECUs and the variants across the ECUs with respect to the current generation E/E architecture to reduce cost and complexity as well as to increase performance.
BN_T7.6_02	Architecture evaluation methods	Investigation of Systematic evaluation methods to compare alternative E/E architectures, including embedded network communications, to identify the most suitable one that meets a given set of requirements as well as supports scalability, flexibility, reusability and adaptability.
BN_T7.6_03	Partitioning and allocation	Exploration of concepts, methods and tools for efficient partitioning and allocation of functionalities as well as software across the multicore ECUs at the E/E architecture level and across the processor cores within a particular ECU to maximize performance and resource utilization.

BN_T7.6_04	Dependability enhancement	Enhancement of dependability (functional safety, reliability, fault tolerance, security, and availability) both at the E/E architecture level and at the ECU level by utilization of available resources of multicores in order to improve product quality and safety.
BN_T7.6_05	Mixed criticality support	Ensure separation, i.e. freedom from interference among applications with different criticality levels (as defined in ISO 26262) and manage shared resources of multicore-based ECUs.
BN_T7.6_06	Communications	Investigation of concepts for Inter-ECU at the E/E architecture level and intra-ECU communication (including topics network topology, protocol, bandwidth, delay, technology) in multicore based architectures.
BN_T7.6_07	Service-oriented architecture	Viability and applicability of concepts and techniques related to Service-oriented Architecture (SoA) for real-time functional safety-related automotive systems.
BN_T7.6_08	Runtime optimization	Viability and applicability of concepts and techniques related to run-time optimizations for real-time safety-related automotive systems.
BN_T7.6_09	AUTOSAR support for multicore, SoA and runtime optimization.	Recommendations to (a) improve multicore support for mixed-critical systems in future releases of AUTOSAR as well as other relevant standards and specifications, and (b) suggest required changes in AUTOSAR to support SoA and run-time optimizations.
BN_T7.6_10	Mixed OS support	Investigation of concepts for mixed OS support in one and the same ECU.

3. References

[1] AUTOSAR www.autosar.org

4. Abbreviations

Table 8: Abbreviations

Abbreviation	Meaning
ACC	Adaptive Cruise Control
AUTOSAR	Automotive Software Architecture
BN	Business Need
ECU	Electronic Control Unit
EMC^2	Embedded multi-core systems for mixed criticality applications in dynamic and changeable real-time environments
KPI	Key Performance Indicator
µC	Micro-Controller

5. Appendix 1: T7.1 UC ADAS and C2X

5.1 KPIs

Table 9: KPIs for use case "ADAS and C2X"

No	Title	Description	BN relation
KPI_T7.1_01	Reduced Testing and Validation	Reduce the number of hours required to test an ADAS system and the number of different vehicles and situations for which the ADAS system needs to be put to the test.	BN_T7.1_01
KPI_T7.1_02	Enhanced Marketability	The ability fo sell ADAS systems in the market and to maintain and extend them in a cost effective and performant manner shall be enhanced	BN_T7.1_02
KPI_T7.1_03	Increased penetration of MCMC systems	EMC^2 multi-core systems are deployed in commercial systems and deliver benefits across defined axes including power consumption, maintainability and real-time response.	BN_T7.1_03
KPI_T7.1_04	Reduction of distinct in-vehicle platforms	The number of distinct and differentiated platforms in the vehicle is reduced as functionality is combined onto single physical entities by means of EMC^2 functionality.	BN_T7.1_04
KPI_T7.1_05	Maintenance of systemic invariants	Third party audit of ADAS systems from safety, security and privacy perspectives is successful while the systems still deliver their intended benefits.	BN_T7.1_05
KPI_T7.1_06	Retention of legacy investment	Substantial proportions of time-served, mature code are retained despite the migration to an EMC2 compliant system.	BN_T7.1_06
KPI_T7.1_07	Upgradability	Systems built according to EMC^2 principles are cheaper and more reliable to maintain than their non-EMC^2 equivalents.	BN_T7.1_07

5.2 Sub task descriptions

5.2.1 ST7.1.1 UC_Requirements and specification

This subtask contributes the requirements and specification to the overall D7.1 deliverable.

5.3 ST7.1.2 UC_Define system architecture

The use case is evolved around a vehicle-centric approach where new ways of communication allow for interaction with the infrastructure and a backoffice. The figure below gives a high-level representation. The progress within communication technology leads to new ways to use and partition ADAS systems.

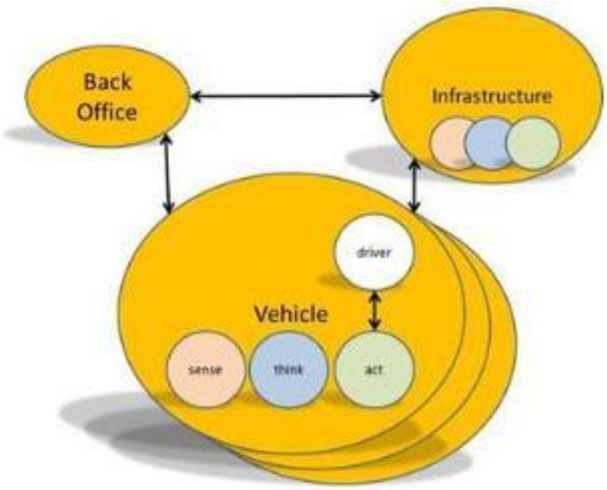


Figure 7: High level representation of system architecture of use case "ADAS and C2X"

Although cellular communication in vehicles is becoming prevalent today, many of the major European OEMs have reached consensus to deliver WLan-P based G5 communication from 2016 (in the context of the Car2Car Communication Consortium). As with any technology migration although penetration will initially be low the technology will become pervasive, leading to new challenges about how to use and rely on V2X data and how to achieve maxiumum Quality of Service in the presence of many vehicles and other wireless communications. The adoption of V2X can further by accelerated by provisioning it into infrastructure elements (to act as virtual sensors) such as at roadworks and at dangerous junctions and by offering after-market units to allow existing deployed fleet to use the V2X facilities. This task studies the implications on the overall system architecture, and delivers a system architecture that leverages in the best way on these anticipated trends.

5.3.1 ST7.1.3 UC_Implementation and integration

Within this task the designed system architecture is implemented. It starts by implementing the different components, of which the majority will be (existing or research-platforms) aftermarket devices. Within the task, also the integration of these components into one system will be delivered. This includes the verification of the different software and hardware modules and the validation of the overall system performance.

The Figure below gives a high-level architecture of the components. It consists of the paradigm "sense, think, act" and can be implemented as such on both the vehicles and the roadside equipment:

- ***Sensing*** is done via the CAN bus (using onboard sensors such the gyroscope, acceleratometers, steering angle and driver actions), camera and radar to monitor the environment, and a communication unit called the ITS router which can be considerd as a "virtual sensor".
- ***Thinking*** is coordinated by the EMC² hardware architecture. Basically it supports a Vehicle State Esimator that observes the dynamic state of the vehicle, an Environmental Perception module that tracks the whereabouts of the vehicle and its perceived context and which provides the Electronic Horizon and ADAS application information.
- ***Acting*** is the part where the vehicle, driver or other elements in the environment are provided with Instructions.

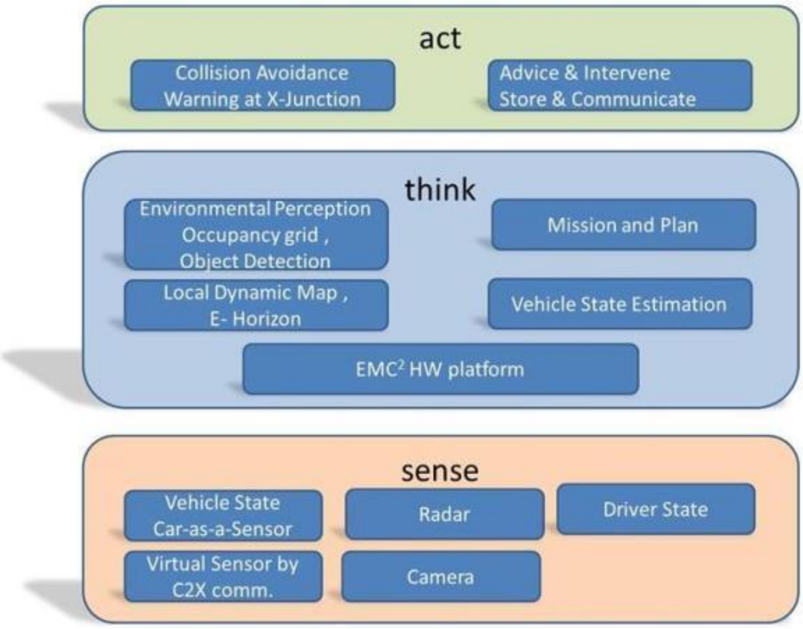

act
Collision Avoidance Warning at X-Junction
Advice & Intervene Store & Communicate
think
Environmental Perception Occupancy grid , Object Detection
Mission and Plan
Local Dynamic Map , E- Horizon
Vehicle State Estimation
EMC² HW platform
sense
Vehicle State Car-as-a-Sensor
Radar
Driver State
Virtual Sensor by C2X comm.
Camera

Figure 8: High-level architecture of use case "ADAS and C2X".

5.3.2 ST7.1.4 UC_Elaborate use cases

This task is responsible for designing and implementing the different use cases.

The rationale to explicitly include infrastructure is as follows. V2I approaches are very similar to V2V approaches (and indeed the two are not mutually exclusive), but have a couple of small advantages; notably infrastructure does not move, which means that once an element of infrastructure has been instrumented (e.g. a junction) then that instrumentation is always available at that location – this allows vehicles to take advantage of it every time they approach, and from considerable distance. This is important since it is reasonable to expect locations where instrumentation will deliver the greatest benefit to be instrumented first; This might be complex junctions, or those with a particular set of issues. The second advantage is that V2I does not suffer from the 'Fax Machine Problem'; Immediately a junction is instrumented a vehicle can reliably take advantage of it.

5.3.3 ST7.1.5 UC_Set-up and execute trials

This task takes care of setting-up and executing different trials to gather hands-on experience. In the Netherlands, the national field operational test (FOT) "Beter Benutten" will start soon. In this FOT a fleet of cars are available for instrumentation and longer term field measurements. Whenever possible, this Living Lab will leverage on this opportunity. In addition, TNO has 5 highly instrumented cars (so-called "carlabs") that are instrumented with synchronized loggers for CAN, RTK-DGPS and accuracte 6DOF motion sensors. One of more of these carlabs will be used to implement the more demanding applications.

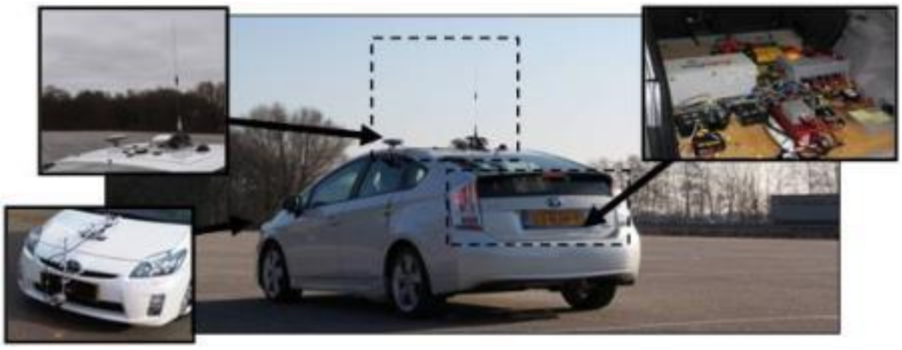


Figure 9: Instrumentation of TNO carlabs vehicle

5.3.4 ST7.1.6 UC_Methodology

This sub-task is concerned with the elements of the migration of ‘traditional’ vehicle soft systems to next generation EMC^2 compliant ones.

Contributions:

1. Safety overview of a generic ADAS functionality
2. Enable migration and integration of existing ADAS sensors and components in multi-core environment
3. Migration of standard software within ADAS use

The task constists of the following activities:

1. Build demonstrator on target platform
2. Manifest based safety specification of software components for reusable software integration
3. Migration from single-core to multi-core keeping previously qualified/defined properties based on results of WP3
4. Safety and reliability as a service based on results of WP3
5. Analytical approach for consolidation of vehicle function on a large scale software integrated system
6. Evaluation for later integration in automotive standards (e.g., AUTOSAR).

5.3.5 ST7.1.7 UC_Evaluation

This task contributes to the overall D7.2 deliverable and represents the assessment of the contribution of the work.

5.4 High level requirements

Table 10: High level requirements for use case “ADAS and C2X”

Requirement ID	Short description	Description	Rationale
HL-REQ-WP07-035	Load and execute program at run time	It shall be possible to load and un-load a software unit during run-time of the system.	Two possible business needs: 1) optimized resource use for applications with disjunct modes of execution. 2) in case of permanent HW defect, re-load application on different core/µC
HL-REQ-WP07-036	WCET Analysability	It shall be possible to give a tight worst case execution time estimate on multi-core hardware (either COTS or soon to be manufactured by project partners).	Required to validate correct execution on multi-core.
HL-REQ-WP07-037	WCET Tightness	It shall be possible for a software unit to request execution with a deadline that can be guaranteed if available at system configuration time.	Separation into realtime and non-realtime allowes optimized use of computation resource.

HL-REQ-WP07-039	It shall be possible to request specific communication to be secured.	It shall be possible to request specific communication to be secured. The secure communication shall have configurable key strength, include authorization, authentication, liveness, encryption, integrity, etc. on demand. Symmetric keys and PKIs shall be supported.	Pre-condition for safety critical applications involving V2X communication. Increasing importance to guarantee integrity of vehicle operation.
HL-REQ-WP07-040	safety on demand	Spatial and temporal protection (e.g., reserve own core own memory with MPU protection) shall be possible on request (at configuration time). Redundant calculations (eiter in parallel or in sequence) with comparison shall be automated on request.	Optimized use of resource. E.g., safety critical part that requires only 10% of computation time can lock two cores on demand. Remaining time free for parallel computation on two cores.
HL-REQ-WP07-042	Minimum size for the executables	The code size of the EMC² system software shall be comparable to the range of current AUTOSAR implementations.	Necessary for acceptance of EMC² technology in automotive industry.
HL-REQ-WP07-043	Minimum memory consumption	The RAM consumption of the EMC² system software and applications shall be comparable to the range of current AUTOSAR implementations.	Necessary for acceptance of EMC² technology in automotive industry.
HL-REQ-WP07-060	The jitter of time triggered tasks	The jitter of time triggered tasks and the latency of event triggered interrupts shall be << 1ms.	Necessary for acceptance of EMC² technology in automotive industry.
HL-REQ-WP07-061	Tool support	A tool framework for application modeling, code generation, verification, and test should be available for the EMC² platforms.	Prevent inconsistencies by one integrated tool framework.
HL-REQ-WP07-062 HL-REQ-WP07-063 HL-REQ-WP07-064 HL-REQ-	project milestones and progress	Initial platform hardware and development tools shall be available to UC1 at MS1 (Nov 2014). A first set of EMC² features shall be implemented in the platform and tools and available to UC1 at MS4 (Aug 2015). The complete setof EMC² features shall be implemented in the platform and tools and available to UC1 at MS7 (Sep 2016).	EMC² technology can only be evaluated in the Use Case, if it is available in time to be implemented and tested. Late deliveries will not be considered and not deployed.

WP07-065 HL-REQ-WP07-066 HL-REQ-WP07-067			

5.5 Evaluation plans

The evaluation of the performance of the use case shall be based upon the way in which the developments demonstrated address the identified business needs (and, thus, by extension, the Requirements). Scoring will be according to how the achieved result compared to the expected result;

A. **Exceeds expectations**

The solution given goes beyond the expectations;

B. **Totally fulfilled**

The solution completely fulfils the expectations;

C. **Partially fulfilled**

The solution given does bring something new but partial covers the expectations;

D. **Not fulfilled**

The solution although attempted, does not fulfill the expectations;

E. **Not covered**

There was no attempt to implement a solution that covers a specific need.

Table 11: Evaluation plans for the high level requirements for use case “ADAS and C2X”

Requirement ID	Short Description	Evaluation Plans
HL-REQ-WP07-035	Load and execute program at run-time	Add and remove units of functionality (defined to be code loads, not configuration parameterisation) from the system while it remains operational and in full function.
HL-REQ-WP07-036	WCET Analysability	Predict the worst case execution time from a case theoretic basis and then compare with the observed performance of the system in a statistically significant manner.
HL-REQ-WP07-037	WCET Tightness	Request execution within a timing constraint and then verify that this constraint was met during execution, in a statistically significant manner.
HL-REQ-WP07-039	It shall be possible to request specific communication to be secured.	Verification by an agency independent of the implementer that the chosen communication infrastructure and supporting implementation meets the identified requirements.
HL-REQ-WP07-040	Safety on demand	Verification by an agency independent of the implementer that the chosen communication infrastructure and supporting implementation meets the identified requirements.
HL-REQ-WP07-042	Minimum size for the executables	Create ‘menu’ of typical AUTOSAR functions and then compare code size for implemented EMC^2 variants with these results.
HL-REQ-WP07-043	Minimum memory consumption	Create ‘menu’ of typical AUTOSAR functions and then compare RAM requirements for implemented EMC^2 variants with these results.

HL-REQ-WP07-060	The jitter of time-triggered tasks	Evaluate jitter performance of implemented functions in a statistically significant manner and assess the results.
HL-REQ-WP07-061	Tool support	By observation.
HL-REQ-WP07-062 HL-REQ-WP07-063 HL-REQ-WP07-064 HL-REQ-WP07-065 HL-REQ-WP07-066 HL-REQ-WP07-067	Project milestones and progress	By project reporting.

6. Appendix 2: T7.2 UC Highly automated driving

6.1 KPIs

Table 12: KPIs for use case "Highly automated driving"

No	Title	Description	BN relation
KPI_T7.2_01	Increased coverage and configurability by simulation	Reduce cost and time for each system design (or adaptation) by at least 10%, while increasing 10% performance and energy efficiency.	BN_T7.2_01
KPI_T7.2_02	Decrease development time by harmonization of development tools	Reduce development time of 10% with respect to the current tool boundaries.	BN_T7.2_02
KPI_T7.2_03	Definition of methodologies to increase confidence in HW/SW integration	Defined methodologies/guidelines to efficiently integrate existing high-level functionalities under a mixed-criticality environment in multicore platforms	BN_T7.2_03
KPI_T7.2_04	Enhance the resources allocation	Increase by at least 20% the level of performances of the different sub-systems by a time-critical dynamic allocation mechanism.	BN_T7.2_04
KPI_T7.2_05	Definition of methodologies to characterize effective portability of SW	Defined methodologies/guidelines to characterize the suitability, constraints, flexibility and reusability of SW for specific HW configurations.	BN_T7.2_05
KPI_T7.2_06	Definition of tools to parallelize and classify SW susceptible to be accelerated	Defined concepts/methods/tools to semi-automatically determine which components or algorithms are susceptible to be accelerated and quantify the resultant performance gains.	BN_T7.2_06

6.2 Sub task descriptions

Figure 1 shows the tasks within this UC and the relation of each task both with other tasks within this UC and with other technical WPs. As can be seen in the figure, subtasks ST7.2.3 – ST7.2.5 are rather independent and mastered by subtask ST7.2.3. These four tasks are expected to be carried out following the requirements and evaluation plans defined in subtask ST7.2.1, and thereafter integrated and evaluated in ST7.2.6. The scope and more specific description of the use case are defined within this task.

Requirements are derived and collected in ST7.6.1 along with definitions of use case specifications, evaluation criteria and evaluation methodology. The requirements are communicated to the relevant tasks of this UC, Automotive Living Lab (WP7), and the other technical WPs (WP1: SP_SoA – Embedded Systems Architecture, WP2: SP_Application Programmability and Static/Offline System Software, WP3: SP_Dynamic Runtime Environments and Services, WP5: SP_System Design Platform. Tools, Models and Interoperability, and in a more perifheral manner WP6: SP_System Qualification and Certification) to perform the required technical activities.

General concepts, techniques and tools are developed in the respective technical WPs. Concepts, methods and tools developed in other WPs are to be customized to meet the requirements and specifications of the use case. Note also that since there are many links among most of the Automotive Living Lab Use Cases, synergies will be exploited to maximize the workflow between them. In particular, Use Case 2 and Use Case 1 share many functional requirements, so a strong collaboration between them will be pursued.

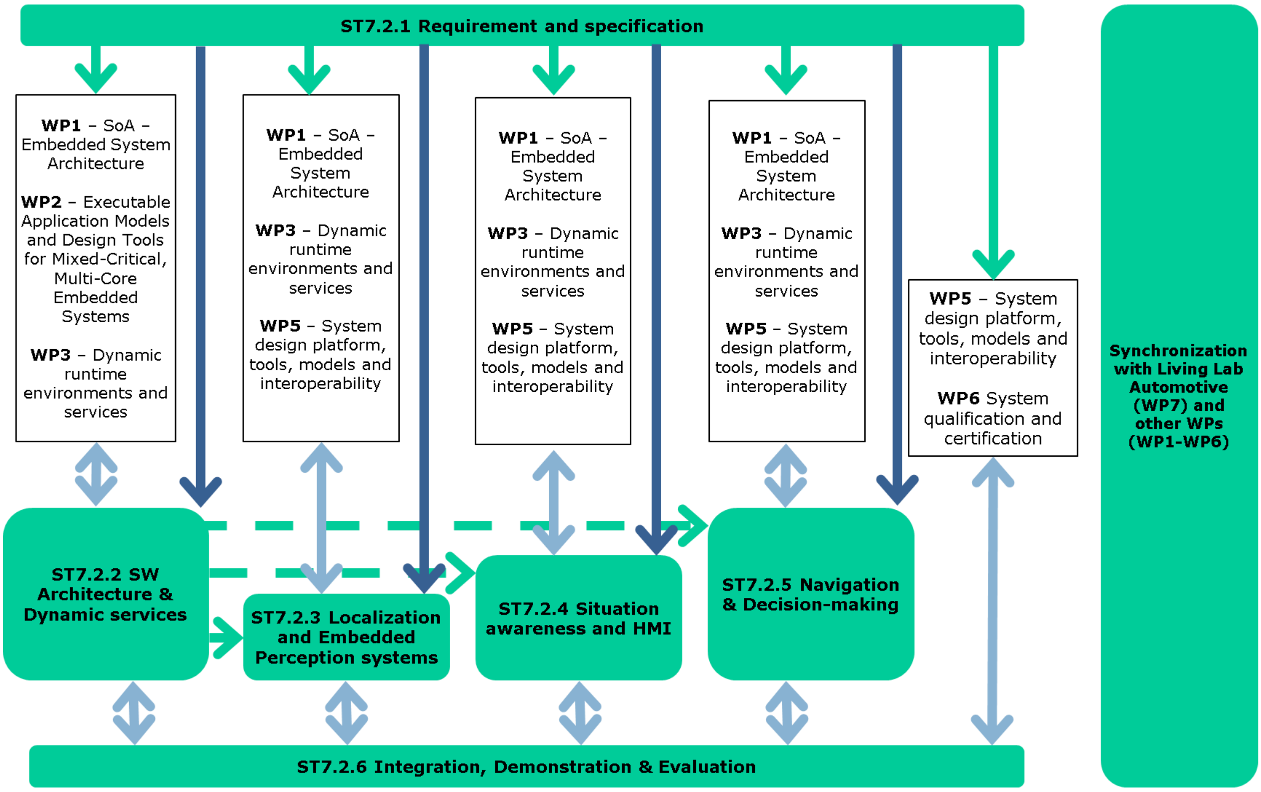

ST7.2.1 Requirement and specification
WP1 – SoA – Embedded System Architecture
WP2 – Executable Application Models and Design Tools for Mixed-Critical, Multi-Core Embedded Systems
WP3 – Dynamic runtime environments and services
WP1 – SoA – Embedded System Architecture
WP3 – Dynamic runtime environments and services
WP5 – System design platform, tools, models and interoperability
WP1 – SoA – Embedded System Architecture
WP3 – Dynamic runtime environments and services
WP5 – System design platform, tools, models and interoperability
WP1 – SoA – Embedded System Architecture
WP3 – Dynamic runtime environments and services
WP5 – System design platform, tools, models and interoperability
WP5 – System design platform, tools, models and interoperability
WP6 System qualification and certification
Synchronization with Living Lab Automotive (WP7) and other WPs (WP1-WP6)
ST7.2.2 SW Architecture & Dynamic services
ST7.2.3 Localization and Embedded Perception systems
ST7.2.4 Situation awareness and HMI
ST7.2.5 Navigation & Decision-making
ST7.2.6 Integration, Demonstration & Evaluation

Figure 10: Subtasks of T7.2 and interaction with WP7 and technical WPs (1-6)

6.2.1 ST7.1.1 Requirement and specification

The expected activities within this sub-task are as follows:

- Low level requirements and evaluation plans will be derived from a first definition of business needs and key performance indicators.
- Specification of a workflow to properly integrate developments or tools provided by the partners of UC2, or other related other UC and WP. It will be mainly based on an iterative development from concept definition to validation.
- Identification of the available or in progress tools and methodologies within the consortium that fulfill the workflow requirements. This task should permit to identify simulation, modelling and validation tools, HW specification, and SW developement methodologies.
- Description of the Use Case goals, the demonstrators to be implemented, and the corresponding evaluation plans. The validation phase will be accomplished according to a set of criteria and metrics also defined in the low-level requirement specification.
- Subsystems technical specifications to properly coordinate the work between Use Case partners and other relevant partners from the automotive Living Lab and the technical WPs.

This sub-task contributes to the common WP7 deliverable (D7.1), and common T7.2 deliverables (D7.5 and D7.6)

Relation to business needs: BN_WP7_T7.2_01
Relation to WPs: WP1 and other Use Cases of the Automotive Living Labs

6.2.2 ST7.2.2 Architecture & Dynamic services

In this sub-task, a dynamic mixed-criticality on board system architecture will be defined in order to guarantee high availability and reliability, taking into considerations local and remote sensors, processing and actuation elements. This task will also use modeling and simulation tools (WP2-WP5) for efficient HW/SW co-design of the embedded multi-core system. Dynamic scheduling approaches will be evaluated in order to balance performance, flexibility and resource efficiency. The specific activities to be performed within this subtask are the following:

- Model behaviors of hierarchical, compositional, and executable SW components within EMC2 SOA architecture in a multi-core target.
- Explicitly consider separation aspects among applications, with different criticalities and management of shared resources.
- Investigate a solution with an increased integration of both the existing functionalities and the improvement/refinements to be developed.
- Explore different strategies for dynamic scheduling and synchronization across the hybrid multi-core computing platform.

This sub-task contributes to the common WP7 deliverable (D7.2), common T7.2 deliverables (D7.5 and D7.6) and an internal report related to the SW Architecture.

Relation to business needs: BN_WP7_T7.2_01, BN_WP7_T7.2_02, BN_WP7_T7.2_04
Relation to WPs: WP1- WP3 and other Use Cases of the Automotive Living Labs

6.2.3 ST7.2.3 Localization and Embedded Perception systems

This sub-task will involve the study and the development of the localization and standalone/cooperative perception systems embedded in the highly automated vehicle. Such systems will use Commercial Off-the-Shelf (COTS) sensors taking into account automotive integration constraints. The solution to be developed will explore the best trade-off performance/computational cost for the specified HW target. More specifically, the following activities will be performed:

- Develop solutions that fuse information from the vehicle proprioceptive CAN-bus automotive sensors, geometrical features (road curvature and width) extracted by low-cost perception sensors and standalone low cost automotive-type GNSS receivers and Inertial Measurement Units;
- Implement reliable systems that perform large-field of view perception with COTS cameras and lidars/radars, being the detected objects static and linked to the infrastructure (traffic lights and signs) or moving (vehicles, pedestrians) on the drivable space. An investigation on the number, configuration and nature of these sensors will be performed to determine the most suitable architecture for the selected HW target on the EMC2 SW architecture basis.
- Integrate high-level information (position, speed) from surrounding vehicles received through V2X communication routers.
- Specify the navigation map characteristics and its attributes to properly use it for localization and situation awareness purposes in the embedded solution.

This sub-task contributes to the common WP7 deliverable (D7.2), common T7.2 deliverables (D7.5 and D7.6) and an internal report related to the Localization and Embedded Perception systems.

Relation to business needs: BN_WP7_T7.2_02, BN_WP7_T7.2_05, BN_WP7_T7.2_06

Relation to WPs: WP1. WP3, WP5 and other Use Cases of the Automotive Living Labs

6.2.4 ST7.2.4 Situation awareness and HMI

In this sub-task, an overall dynamic local map will be generated from the onboard perceived entities and those received from the V2X communications. It will be the basis to derive a risk assessment considering estimated intentions and expectations of other road users, and for further decision-making. In the final step of the information flow, the driving situation interpretation will be properly shown to the driver in order to increase its situation awareness in case of take-over. This objective can be decomposed in the following activities

- Evaluation of resources allocation mechnisms for the generation of a Probabilistic Dynamic Local Map where several graceful degradation modes are to be considered.
- Exploitation of parallelization possibilities for risk assessment approaches based on Bayesian Dynamic Networks.
- Investigation of the level of affordable complexity for the Human- Machine Interface within the proposed Service-oriented architecture where safety critical tasks (interaction) have to coexist with tasks of a lower level of criticality (3D map representation).

This sub-task contributes to the common WP7 deliverable (D7.2), common T7.2 deliverables (D7.5 and D7.6) and an internal report for Situation awareness systems and HMI.

Relation to business needs: BN_WP7_T7.2_02, BN_WP7_T7.2_05, BN_WP7_T7.2_06
Relation to WPs: WP1. WP3, WP5 and other Use Cases of the Automotive Living Labs

6.2.5 ST7.2.5 Navigation and Decision-making

This sub-task will be in charge of developing the decision system for the supervised automated driving when making complex maneuvers, by taking into account surrounding vehicles and infrastructure information. From the information provided by onboard localization and perception systems (ST7.2.2), and the risk assessment of the scene (ST7.2.3), the implemented decision system will be able to plan and to execute the safest and more efficient strategy within a human-like driving context. In addition, advanced services based on vehicles connectivity like Dynamic ridesharing (DR) and Intelligent Parking Management (IPM) will be evaluated in conjunction with the highly automated driving experience. The main activities to be pursued within the sub-task are the following

- Introduce reactive Motion Planning considering uncertainties to take advantage of the Bayesian framework used all through the data flow of the subsystems developed in ST7.2.3 and ST7.2.3.
- Evaluate mixed criticalities handling by running self-Optimization and On-line Learning tasks within the same functional component in charge of the Decision System, which is intrinsically time-critical.
- Investigate emergency human-in-the-loop control strategies to properly allow the transition between human driving and automated driving
- Implement through a traffic simulator DR and IPM functionalities and evaluate its proper integration with the embedded navigation system and the HMI.

This sub-task contributes to the common WP7 deliverable (D7.2), common T7.2 deliverables (D7.5 and D7.6) and an internal report describing Navigation and Decision-making systems.

Relation to business needs: BN_WP7_T7.2_02, BN_WP7_T7.2_05, BN_WP7_T7.2_06

Relation to WPs: WP1. WP3, WP5 and other Use Cases of the Automotive Living Labs

6.2.6 ST7.2.6 Integration, Demonstration & Evaluation

This sub-task takes will deal with systems integration, the setting-up and execution of the selected demonstration means: both simulating the embedded system in an ADAS environment simulator where HIL is possible, and in different trials in the selected test-site with real vehicles. IXION has access to 3/4 computer-controlled vehicles that will be put together to show the suitability of Highly Automated Driving Use Case in an urban scenario (private facilities), with access to current infrastructure elements (traffic lights) and RSU sensors (camera).
The validation and optimization of the architectural design and system behavior in the selected scenarios will be also assessed at this stage, as previously stated both in co-simulation and in experimentation. Special attention will be given to dependability and safety. The main activities to v b developed are:

- Integration of techniques and tools developed or customized within this use case (or in related WPS) to facilitate demonstration and evaluation of the use case
- Iterative SW/HW integration within EMC2 SoA architecture of the developments performed in ST7.2.2- ST7.2.5
- Test-site preparation to properly demonstrate on a real platform the goals and expected results of the Use Case. Two demonstrators are foreseen in M24 and M36 with different degrees of completion
- Technical and functional validation according to evaluation criteria, metrics and methods defined in ST7.2.1
- Provide inputs to future releases of relevant standards and specifications (in particular ISO 26262 and AUTOSAR)

This task will contribute to the common WP7 deliverable (D7.2), common T7.2 deliverable (D7.6), and one internal report regarding systems integration, use case demonstration and evaluation results.

Relation to business needs: BN_WP7_T7.2_01, BN_WP7_T7.2_02
Relation to WPs: WP5, WP6 and other Use Cases of the Automotive Living Labs

6.3 High level requirements

Table 13: High level requirements for use case "Highly automated driving"

Requirement ID	Short description	Description	Rationale	BN relation	Sub-task relation
HL-REQ-WP07-001	Modeling and simulation environment	Enhancement of simulation environments to properly evaluate High Level functionalities within the adopted multicore SOA architecture, integrating performance and power modeling capabilities.		BN_T7.2_01	ST7.2.1, ST7.2.2, ST7.2.6
HL-REQ-WP07-002	Resource optimization	Dynamic optimization of computational resources, hence energy consumption and costs, according to the performance requirements without putting quality at		BN_T7.2_02	ST7.2.2, ST7.2.3, ST7.2.4, ST.2.5

		risk.			
HL-REQ-WP07-003	Parallelization and classification	Development of functionalities able to (i) recognize computing structures of application source code that may be suitable for parallel execution and (ii) to classify and evaluate different parts of the algorithms and decide if they can be effectively moved to any accelerator component of the current HW.		BN_T7.2_03	ST7.2.6
HL-REQ-WP07-004	Effective portability	Characterization of software components on one hardware platform so that they can be located on different hardware platforms without running the full set of timing and analysis on each platform; this supports a higher flexibility in the design of the system		BN_T7.2_04	ST7.2.2
HL-REQ-WP07-005	Harmonization of development tools	Availability of a set of harmonized tools that allow to easily integrate the design and development work flow, from system to component/code level.		BN_T7.2_05	ST7.2.3, ST7.2.4, ST7.2.5
HL-REQ-WP07-006	Integration in heterogeneous platforms	Exploration of methods and tools able to increase confidence (reduce risk) of problems being observed at integration time, taking into consideration all SW layers.		BN_T7.2_06	ST7.2.3, ST7.2.4, ST7.2.5

6.4 Evaluation plans

Table 14: Evaluation plans for the high level requirements for use case "Highly automated driving"

Requirement ID	Short description	Evaluation plans
HL-REQ-WP07-001	Modelling and simulation environment	The objective is to explore at what extent high level functionalities can be assessed within a modelling and simulation environment for ADAS, adapted the multicore SOA architecture. To that end: • Some relevant metrics taking into consideration performance (computational cost, required memory, WCET and deadlines fulfilment) and power will be identified • World-time synchronization (bandwidth vs latencies) between traffic and ADAS simulators will be assessed. • Component-based modelling methodology will be thoroughly assessed to quantify the gains in terms of productivity (development

Requirement ID	Short description	Evaluation plans
		cycle) and flexibility (reusability of components) • The behaviour of Hardware-in-the-loop simulation will be compared to SIL in terms of portability, separation and integration.
HL-REQ-WP07-002	Resource optimization	The objective is to investigate dynamic optimization mechanisms of computational resources, according to the performance requirements without putting quality/security at risk. This high level requirement will be evaluated in the light of the following uncertainty considerations • Perception uncertainty, both in terms of deadlines fulfilment (soft-real time components) and of the accuracy of the outputs and its propagation effect in subsequent components (mainly in risk assessment) • Graceful degradation of positioning in complex scenarios, and its consequences in risk estimation and decision-making components • Mixed-criticalities management when perception and localization integrity is low, and its effect on human-in-the-loop decision-making mechanisms
HL-REQ-WP07-003	Parallelization and classification	The objective is to exploit the potential of multicores to enhance the performance of specific components. The evaluation of this requirement will focus on the ability to recognize and classify computing structures of application source code that may be suitable for parallel execution in • occupancy grids for stereo-vision based perception and • sequential importance sampling for risk assessment
HL-REQ-WP07-004	Effective portability	The objective is to investigate at what extent software components can be HW independent, or otherwise, how portable it is, and as a result, how flexible is the code of the developed functionalities. A special focus will be put in timing analysis and memory resources within two differentiated activities: • Analyse graceful degradation of critical components within a mixed criticality setting • Investigate the consistency of ubicuous sensor interfaces (in particular comparing between simulated and real sensors)
HL-REQ-WP07-005	Harmonization of development tools	The objective is to facilitate a seamless integration of the tools for designing, developing and validating components, This requirement will be evaluated at functional level by comparing the real experiments and the simulation HIL environment, including • Implementation of Dynamic Ridesharing and Intelligent Parking Management functionalities within the traffic simulator • Verification of the coherence between the overall Traffic/ADAS simulation environment and the trials with real cars.
HL-REQ-WP07-006	Integration in heterogeneous platforms	To objective is to explore methods and tools able to increase confidence (reduce risk) of problems being observed at integration time: • Investigating how heterogeneous platforms can be properly introduced in the component-based modelling platform • Assessing how the new Hybrid SoCs can be optimized from the HIL simulation platforms • Exploring the potential problems related to the introduction of multiple sensors with high bandwidth (stereo-vision, multiple-layer lidars)

7. Appendix 3 T7.3 UC Design and validation of next generation hybrid powertrain / E-Drive

7.1 KPIs

Table 15: KPIs for use case "Design and validation of next generation hybrid powertrain / E-Drive"

No	Title	Description	Sub-task relation
KPI_T7.3_01	Increased verification coverage by simulation	Increase verification coverage that can be performed in simulation by 10% for highly parallel applications and thus reduce the validation effort on target HW	ST7.3.1, T7.3.5
KPI_T7.3_02	Benefits of a seamless modeling method	Improve design specification maturity and consistency over the different domains, thus reducing cost and time for system design and safety recertification by 15%	ST7.3.2
KPI_T7.3_03	A collection of SW design and development guidelines for multicore platform	Prepare for each SW layer minimum 1 document including guidelines for the design and development of this component on multicore platforms, thus reducing training time for new SW engineers	ST7.3.1, T7.3.3
KPI_T7.3_04	Tailored safety architecture	Defined safety mechanisms for multicore to cover all automotive safety levels	ST7.3.1, ST7.3.2, ST7.3.3
KPI_T7.3_05	Decrease development time by use of integrated toolchain	Reduce development time at tool boundaries by 10% by efficient data transfer and consistency check	ST7.3.1, ST7.3.4, ST7.3.5
KPI_T7.3_06	Safety case generation	Reduction of effort for safety case generation by 20 %	ST7.3.3, ST7.3.4
KPI_T7.3_07	Increased data throughput	Increase data throughputs of existing communication channels by the development and integration of new technologies such as CAN FD, Ethernet	ST7.3.6
KPI_T7.3_08	Data exchange and communication strategies	Maintain same configuration and handling efforts for communication complexity / load increase of 50%	ST7.3.6

7.2 Sub task descriptions

7.2.1 ST7.3.1 UC_Electrical and functional integration of high dynamic e-drive system controls with time based vehicle control algorithm

Multi-core processor systems offer a substantial potential for integrating diverse calculation tasks /calculationalgorithms on one computing unit which are currently executed on two different controller units. This is especially appealing for strongly cost-driven automotive industry as it enables the economization of one embedded controller unit.

The development target of this task therefore is a cost-efficient, robust and reliable runtime environment for electric powertrains in HEV-, PHEV and EV-applications which combines the functionality of an E-Motor Controller Unit (EMCU) with the Vehicle Controller Unit (VCU) in one automotive controller unit (VEMCU). By investigating the capabilities of the multi-core technology in one concrete application a bottom-up approach is followed, as the results acquired will be integrated in generally valid functional design guidelines. Goals of this task are

- to provide requirements and specifications for the application
- to perform / support integration of the different tailored technologies (outcomes from the other tasks)
- to perform evaluation of the proposed EMC² technologies

Relation to business needs: BN_T7.3_01, BN_T7.3_03, BN_T7.3_04, BN_T7.3_05
Relation to WPs: WP1, WP3, WP4

7.2.2 ST7.3.2 UC_Model-driven system / software / hardware / safety engineering for embedded multi core hybrid powertrain and e-Drive control systems

Goal of this task is to enhance model-driven systems engineering approaches in order (1) to better take into account the multi core aspects during system / software / hardware / safety engineering, (2) propose seamless (architecture) modeling approaches to minimize the specification gap between the different disciplines, and (3) improve the links between specification and development tasks. This work covers the following aspects

- Enhancement, tailoring and integration of semi-formal system / architecture description languages (e.g., SysML, CESAR Meta Model, EAST-ADL, AUTOSAR) in order to provide a unified framework for the seamless specification of embedded systems
- Efficient integration of integrated safety mechanisms into an overall concept. This includes the seamless development of the safety concept as well as its verification (dedicated analysis) and validation (test).
- Developing an automotive domain ontology to reveal the semantics of the interaction between different controller units. This ontology will enable integrating mechanisms to compensate unit failures.

Relation to business needs: BN_T7.3_02, BN_T7.3_04
Relation to WPs: WP1, WP3

7.2.3 ST7.3.3 UC_Programming paradigms and SW architecture for embedded multi core hybrid powertrain and e-Drive control systems

Goal of this task is to understand the paradigm shifts while porting an (existing) application from a single core microcontroller to a multi-core platform. The impact of parallelism shall be analyzed w.r.t. complex control algorithms, SW development paradigm and SW architecture (e.g., AUTOSAR), tooling (including compiler) as well as the efficient integration of services the HW platform provide.

Following topics shall be addressed

- Programming paradigms and porting of SW applications to multi-core platforms; Impact regarding programming models, SW architectures, operating systems, task allocation, timing issues and WCET
- Efficient integration of functions and safety into an overall concept. More especially, following aspects shall be focused on:
  - Concept for SW Architecture and Partitioning (re-grouping e.g. VCU and MCU on one silicon), hence also ensure coexistence of mixed-criticality functions and consider also interprocessor communication.
  - Definition and Implementation of scalable safety mechanisms to target applications from ASIL A – ASIL D
  - Definition and Implementation of efficient and safe library for mathematical operations and memory access

Relation to business needs: BN_T7.3_03, BN_T7.3_04, BN_T7.3_06
Relation to WPs: WP2

7.3 High level requirements

Table 16: High level requirements for use case “Design and validation of next generation hybrid powertrain / E-Drive”

Requirement ID	Short description	Description	Rationale	BN relation	Sub-task relation
HL-REQ-WP07-007	Seamless tool chain - tool integration	Industrial tools such as e.g., Enterprise Architect, MKS, Matlab Simulink shall be integrated into a tool chain supporting seamless model-based systems / SW / safety engineering based on a semi-formal system description (e.g., EAST-ADL, SysML)		BN_T7.3_01	ST7.3.1, T7.3.5
HL-REQ-WP07-008	SW architectures - safety mechanisms	Scalable safety mechanisms shall be defined and implemented on the multicore platform to cover all automotive safety levels.		BN_T7.3_02	ST7.3.2
HL-REQ-WP07-009	Safety case generation - automated compilation	The safety case information shall be automatically compiled based on existing information from product development.		BN_T7.3_03	ST7.3.1, T7.3.3
HL-REQ-WP07-010	Simulation environment - multi-core emulation	A simulation environment shall be defined and implemented to provide emulation capabilities for multi-core platforms (highly parallel applications). This shall include following capabilities for an early evaluation of the SW design: a) Multi-rate		BN_T7.3_04	ST7.3.1, ST7.3.2, ST7.3.3

		simulation b) Scheduling analysis			
HL-REQ-WP07-011	Simulation environment - safety analysis	A simulation environment shall be specified and implemented in order to be able to provide support for the ISO26262 defined safety analysis of automotive safety critical applications.		BN_T7.3_05	ST7.3.1, ST7.3.4, ST7.3.5
HL-REQ-WP07-012	Network communication - data exchange	New approaches for the configuration and data exchange between the multicore control unit and developement / test systems over existing automotive communication channels /protocols shall be provided.		BN_T7.3_06	ST7.3.3, ST7.3.4

8. Appendix 4: T7.4 UC Modelling and functional safety analysis of an architecture for ACC system

8.1 KPIs

Table 17: KPIs for use case "Modelling and functional safety analysis of an architecture for ACC system"

No	Title	Description	BN relation
KPI_T7.4_01	Improved modeling and verification of a safety mixed critical system according to ISO 26262 process.	Make effective through a modeling environment the trustworthiness and, finally, the safeness specifications at concept level of ADAS oriented safety mixed critical system (e.g. Adaptive Cruise Control) (starting from the state of the art prototype, the 100% of possible architecture malfunctions and related hazards has to be analyzed in the corresponding operational situations and at least 75% of the necessary safety requirements at concept level have to be defined).	BN_T7.4_01, BN_T7.4_02, BN_T7.4_03
KPI_T7.4_02	Improved ISO 26262 compliant safety case generation.	Reduction of effort for safety case generation by 20 %.	BN_T7.4_02

8.2 Sub task descriptions

The current design of automotive safety critical systems shall be supported by a suitable modelling methodology that could allow the conformity to ISO 26262 standard. This kind of effort should be managed since the beginning of the design, considering the item definition recommendations and requirements from the standard and the subsequent hazard analysis and risk assessment outlined in the process. Then the safety requirements shall be derived according to the standard and verified.
Using a modelling approach by semi-formal language to perform this process, it is important to take care of the general requirements recommended by the standard. The aim is to model the safety critical systems and their safety requirements in order to produce a virtual concept representing the architectural structure and the outcomes of the functional safety analysis, in a potential semi-automated way to be suitable for a future conformity assessment more reliable/robust and less cost/time consuming.

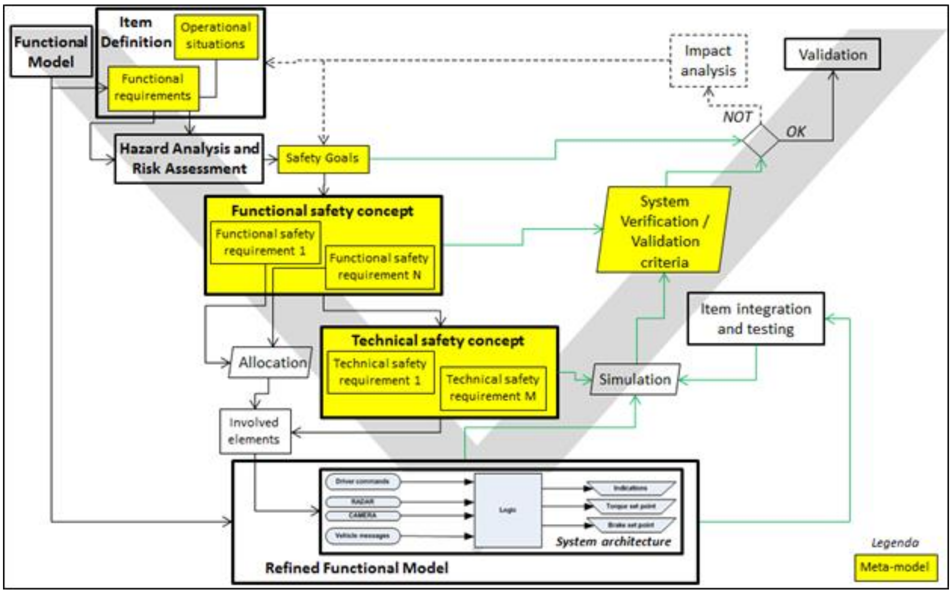


Figure 11: Overall framework for UC7.4

The work is broken down into four work tasks as described in the following.

8.2.1 ST7.4.1 UC_Requirements and specification of an ACC system architecture for advanced vehicles

The task will be devoted to the definition of the requirements and to the specification of ACC system architecture for advanced vehicles derived from the available state of art.

8.2.2 ST7.4.2 UC_Analysis and modelling of an ACC system architecture for advanced vehicles

The focus will be centred on the ACC system that will be considered and modelled from the functional safety perspective in the context of a semi-formal language: SysML inEnterprise Architect will be considered for this aim. Models for this aim have to be defined and designed with their relationships and for requirements allocation in the context of a methodological approach conformant to the ISO 26262 standard process.

8.2.3 ST7.4.3 UC_Functional safety implications according to ISO 26262 for the modeled architecture

On the modelled architecture a hazard analysis and risk assessment will be performed according to the ISO 26262 standard and the functional safety concept will be derived, generating the functional safety requirements necessary for the future implementation and the related criteria for validation. The process will be managed in the context of the semi-formal language elected for the previous sub-task. Specific models for safety requirements will be defined and designed taking care of the ISO 26262 standard prescriptions, related to ASIL inheritance/decomposition and to derivation/traceability of the safety requirements from upper to lower levels.

8.2.4 ST7.4.4 UC_Evaluation

The modelled architecture will be refined according to the functional safety requirements and the safety goals derived from the hazard analysis and risk assessment will be validated according to the validation criteria. The evaluation environment will be again the semi-formal language context and its framework, in which the process has been deployed. The methodology will provide a way for the safety requirements evaluation/testing.

Relation to business needs: BN_WP7_T7.4_01 - BN_WP7_T7.4_03
Relation to WPs: (WP2), WP5, (WP6)

8.3 High level requirements

Table 18: High level requirements for use case "Modelling and functional safety analysis of an architecture for ACC system"

Requirement ID	Short Description	Description	Rationale	BN relation	Sub-Task relation
HL-REQ-WP07-014	System architecture and functional requirements modeling in semi-formal language (e.g. UML/SysML) from Item Definition according to ISO 26262.	The avaialble characteristics and functionalities, taking into account also the mixed criticality, from the product description should be modeled in semi-formal language (e.g. UML /SysML) within a suitable environment/framework (e.g. Enterprise Architect), in compliance with the Item Definition guidelines according to ISO 26262 standard.	Need to represent in semi-formal language (e.g. UML/SysML) within a modeling framework environment (e.g. Enterprise Architect) the architecture and the functional requirements of a safety mixed critical system, for supporting its structured functional safety analysis in compliance with ISO 26262 standard.	BN_T7.4_01, BN_T7.4_02, BN_T7.4_03	ST7.4.1, ST7.4.2
HL-REQ-WP07-015	Bi-directional translation and link between formal structure (e.g. Simulink) modeling and semi-formal (e.g. UML/SysML) modeling of the system architectures.	The available models of the system architecture in a formal modeling/simulation environment (e.g. Simulink) should be translated and linked to a semi-formal modeling/simulation environment (e.g. UML/SysML in Enterprise Architect) and viceversa.	Need to maintain a link between an operating/executable simulation environment (e.g. Simulink) and a semi-formal modeling environment (UML/SysML) that supports the process development of the functional safety assessment according to ISO 26262 standard.	BN_T7.4_01, BN_T7.4_02, BN_T7.4_03	ST7.4.1, ST7.4.2
HL-REQ-WP07-016	ISO 26262 Hazard Analysis and Risk Assessment results translation/modeling in semi-formal language/environment (e.g. UML/SysML in Enterprise Architect).	ISO 26262 Hazard Analysis and Risk Assessment starts from the functional requirements and operational situations from the Item Definition and leads to the safety goals, with safe states, definition as the top level safety requirements. The modeling environment in semi-formal language (e.g. UML/SysML in Enterprise Architect) can encompasse the results of this step of ISO 26262 process.	This is the transition step of ISO 26262 standard process from the Item Definition and the specification of the safety requirements, according to ISO 26262 standard.	BN_T7.4_01, BN_T7.4_02, BN_T7.4_03	ST7.4.2

HL-REQ-WP07-017	ISO 26262 safety requirements chain definition and derivation (starting from the safety goals) modeled in semi-formal language (e.g. UML /SysML).	From the safety goals the safety requirements chain is derived (functional safety req., tecnical safety req, HW safety req., SW safety req., design spec.) and modeled in a semi-formal language (e.g. UML/SysML), taking into account also the mixed criticality, according to the safety requirements specifications prescribed by ISO 26262 standard.	This is the part of ISO 26262 standard process in which the safety requirements are produced. The semi-formal language (e.g. UML/SysML) supports a structural deployment of this phase.	BN_T7.4_01, BN_T7.4_02, BN_T7.4_03	ST7.4.3
HL-REQ-WP07-018	ISO 26262 safety requirements allocation to the system architecture (simulink and semi-formal language).	According to the ISO 26262 standard the safety requirements should be allocated to the elements of the system architecture. Links to these elements should be established through the modeling environment in semi-formal (e.g. UML/SysML) e the formal (e.g. Simulink) languages.	Each safety requirement has to be allocated to a system architecture element according to ISO 26262 standard specification.	BN_T7.4_01, BN_T7.4_02, BN_T7.4_03	ST7.4.3
HL-REQ-WP07-019	ISO 26262 safety requirements internal compliance demonstration from lower level requirements to higher level requirements.	Each safety requirement derived and modeled should be demostrated compliant towards its predecessor at upper level, according to ISO 26262 specification, provided a suitable mean of automatic demonstration of this coherence.	Each safety requirement has to be compliant with its predecessor according to ISO 26262 standard specification.	BN_T7.4_01, BN_T7.4_02, BN_T7.4_03	ST7.4.4
HL-REQ-WP07-020	ISO 26262 safety requirements verification tests automatic generation.	Automatic generation of test patterns and their execution for each safety requirement, according to ISO 26262 standard.	Each level of safety requirements should be verified through a set of testing specification/criteria, according to ISO 26262 standard specification.	BN_T7.4_01, BN_T7.4_02, BN_T7.4_03	ST7.4.4
HL-REQ-WP07-021	ISO 26262 safety requirements traceability management.	The safety requirements modeled in the semi-formal language (e.g. UML/SysML) should be traceable according to ISO 26262 standard.	Management of the change impact through the entire safety requirements chain.	BN_T7.4_01, BN_T7.4_02, BN_T7.4_03	ST7.4.4
HL-REQ-WP07-022	ISO 26262 safety case (before integration/validation) automatic generation from the results consequent to the above requirements.	The resutls available from the previous steps can be automatically collected and linked for the assembly of the safety case (before integration/validation).	The steps above described will produce the main content of the safety case (before integration/validation) accordingly to ISO 26262 standard process, supporting the functional safety assessment.	BN_T7.4_02, BN_T7.4_03	ST7.4.4

8.4 Evaluation plans

Several elements could be evaluated, depending of the maturity of the available methodology.

The evaluation can start from the effectiveness of the links (e.g. translation) of SysML modeling into a Simulink environment in order to constitute a base for the testing phase.

Then the requirements traceability has to be maintained overall their entire chain, until to the verification phase with respect to Simulink environment, provided the suitable models for testing, and a feedback should be provided, after the testing, versus the tested safety requirements. The Simulink context can support the evaluation of the methodology developmed in SysML, offering the base for verifying the modeling of simulation tests.

Table 19: Evaluation plans for use case “Infotainment and eCall Application Multi-Critical Application”

Requirement ID	Short Description	Evaluation plans
HL-REQ-WP07-014	System architecture and functional requirements modeling in semi-formal language (e.g. UML/SysML) from Item Definition according to ISO 26262.	Evaluation of modeling effectiveness in terms of system architecture levels and their traceability (functional versus technical) and safety requirements allocation (see next HL-REQ-WP07-018)
HL-REQ-WP07-015	Bi-directional translation and link between formal structure (e.g. Simulink) modeling and semi-formal (e.g. UML/SysML) modeling of the system architectures.	Evaluation of modeling translation capability considering the previous requirement.
HL-REQ-WP07-016	ISO 26262 Hazard Analysis and Risk Assessment results translation/modeling in semi-formal language/environment (e.g. UML/SysML in Enterprise Architect).	Capability of defining suitable artifacts for representing hazards and hazardous events versus safety goals and maintaining the traceability.
HL-REQ-WP07-017	ISO 26262 safety requirements chain definition and derivation (starting from the safety goals) modeled in semi-formal language (e.g. UML /SysML).	Capability of defining suitable artifacts for representing safety requiremenmts at all levels and maintaining their traceability.
HL-REQ-WP07-018	ISO 26262 safety requirements allocation to the system architecture (simulink and semi-formal language).	Capability in relation to the previous HL-REQ-WP07-014 and HL-REQ-WP07-015.
HL-REQ-WP07-019	ISO 26262 safety requirements internal compliance demonstration from lower level requirements to higher level requirements.	Capability of modeling/managing requirement versus their reciprocal coherence.
HL-REQ-WP07-020	ISO 26262 safety requirements verification tests automatic generation.	Capability of generating tests from safety requirements and system architectures modeling versus code generation and/or Simulink environment.
HL-REQ-WP07-021	ISO 26262 safety requirements traceability management.	Realted to previous HL-REQ-WP07-014, HL-REQ-WP07-015 and HL-REQ-WP07-019.
HL-REQ-WP07-022	ISO 26262 safety case (before integration/validation) automatic generation from the results consequent to the above requirements.	Capability of generating a summary of reporting from previous requirements.

9. Appendix 5: T7.5 UC Infotainment and eCall Application Multi-Critical Application

9.1 KPIs

Table 20: KPIs for use case "Infotainment and eCall Application Multi-Critical Application"

No	Title	Description	BN relation
KPI_T7.5_01	Mixed criticality tasks	The execution environment must support the deployment of tasks with at least two distinct levels of criticality	ST7.5.2
KPI_T7.5_02	Multicore tasks	The execution environment must support the deployment of concurrent multi-core applications.	ST7.5.2, ST7.5.3, ST7.5.4
KPI_T7.5_03	Secure communication	A communication mechanism must be available for secure data transfer between two environments of distinct criticalities	ST7.5.2, ST7.5.3, ST7.5.4
KPI_T7.5_04	Android	The Android Operating System shall be the runtime environment used as the Infotainment system	ST7.5.5
KPI_T7.5_05	FreeRTOS	The FreeRTOS shall be used as the runtime environment for the RTOS	ST7.5.5
KPI_T7.5_06	Infotainment	The infotainment system shall use the Android applications for its multimedia purposes	ST7.5.3, ST7.5.4, ST7.5.5
KPI_T7.5_07	System devices	HW resources shall be managed by their assigned OS	ST7.5.5
KPI_T7.5_08	eCall system	An eCall system shall be developed as a critical real time task on the RTOS runtime environment	ST7.5.4, ST7.5.5
KPI_T7.5_09	Software Containment	The distinct levels of criticality shall be addressed using hardware assisted virtualization.	ST7.5.2, ST7.5.3, ST7.5.4, ST7.5.5
KPI_T7.5_10	Resource partitioning	The execution environment shall allow for space and time partitioning.	ST7.5.2, ST7.5.3, ST7.5.4, ST7.5.5
KPI_T7.5_11	Resource takeover	The execution environment shall support non-critical resource takeover by critical component initially determined as non-critical.	ST7.5.2, ST7.5.3, ST7.5.4, ST7.5.5
KPI_T7.5_12	Inter-core communication	The execution environment shall support an inter-core communication API and be able to send a signal between cores.	ST7.5.3, ST7.5.4, ST7.5.5
KPI_T7.5_13	Execution monitoring	The execution environment should at least support the monitoring of CPU utilization and idle time.	ST7.5.5
KPI_T7.5_14	Core affinity	The execution environment should allow binding a function to a specific core.	ST7.5.5

9.2 Sub task descriptions

In order to develop a multi-core multi-criticality demonstrator we've split into sub-tasks the various steps required to design, develop and integrate, technology that shall be the use case's demonstrator.

As described in Figure 12, the use case shall start with the collection of requirements in ST7.5.1 that will propagate to both the use case sub-tasks and to the WP3 technology tasks. In ST7.5.2 the security aspects shall be addressed throughout the duration of the use case task with tight integration with the T3.5. Technologies developed in WP3 shall then later be integrated into the use case demonstrator within ST7.5.3 while the infotainment and eCall applications are within the ST7.5.4. All these sub-tasks shall then be finalized during ST7.5.5 where all components shall be fine tuned to conclude the use case demonstrator, which will be then reviewed in ST7.5.6.

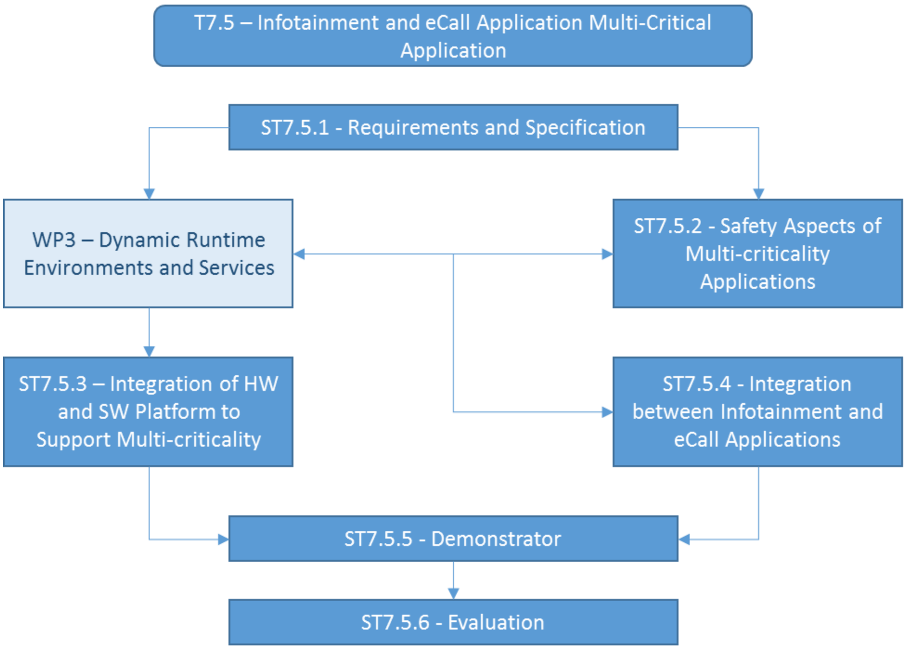

T7.5 – Infotainment and eCall Application Multi-Critical Application
ST7.5.1 - Requirements and Specification
WP3 – Dynamic Runtime Environments and Services
ST7.5.2 - Safety Aspects of Multi-criticality Applications
ST7.5.3 – Integration of HW and SW Platform to Support Multi-criticality
ST7.5.4 - Integration between Infotainment and eCall Applications
ST7.5.5 - Demonstrator
ST7.5.6 - Evaluation

Figure 12: Use case sub-tasks workflow

9.2.1 ST7.5. Requirements and specification

In this sub-task the effort will be focused on the user and system requirements applicable to the use case based on the market needs. These requirements would be the basis for the use case demonstrator, as well to the WP3 tasks where some of the required components shall be implemented.

9.2.2 ST7.5.2 Safety aspects of multi-criticality applications

To ensure the required safety aspects of the demonstrator we must take into consideration at first, still the design level, measures suchs as fault injection and runtime monitoring coupled to the demonstrator components actively or passively ensuring the platform safety, as well the measures the platform must take in case of a fail condition.

9.2.3 ST7.5.3 Integration of HW and SW platform to support multi-criticality

This sub-task is intended to merge the technology developed in WP3 as separate components and integrate them into the use case demonstrator hardware.

)

9.2.4 ST7.5.4 Integration between infotainment and eCall applications

Here the effort shall be used to integrate the infotainment platform and the eCall application to the demonstrator platform. Each has its distinct criticality, timming and safety needs. As such, while running concurrently on the same hardware, one must be run isolated from the other.

9.2.5 ST7.5.5 Demonstrator

This shall be the last development sub-task where each aspect of the use case demonstrator shall be fine tuned to provide the final version of the use case demonstrator.

9.2.6 ST7.5.6 Evaluation

In this final sub-task, the consortium shall evaluate the effectiveness of the technology developed in this use case.

9.3 High level requirements

Table 21: High level requirements for use case "Infotainment and eCall Application Multi-Critical Application"

Requirement ID	Short Description	Description	Rationale	BN relation	Sub-Task relation
HL-REQ-WP07-033	Mixed criticality tasks	The execution environment must support the deployment of tasks with at least two distinct levels of criticality	Mixed-criticality demand comes with the need of supplying enhanced features without increasing certification costs. In order to comply with safety standards, software components interacting with safety features require a level of certification that would otherwise become too expensive for feature rich runtime environments.	BN_T7.5_01	ST7.5.2 ST7.5.3 ST7.5.4 ST7.5.5
HL-REQ-WP07-034	Multicore tasks	The execution environment must support the deployment of concurrent multi-core applications.	Another issue with ever increasing features is the need of failure partitioning, where a feature crashing must be contained within safety boundries. This is typicaly solved by duplicating hardware for each safety impact a feature requires. For example the cars ESP system and low pressure tire detection system have distinct safety impacts if a software/hardware crash would to happen, so although colecting similar information (tire's rotation) they're physically partitioned. A	BN_T7.5_02	ST7.5.2 ST7.5.3 ST7.5.4

			multi-core system with mixed-criticality would then be able to execute both concurrently on the same hardware platform while still partitioned.		
HL-REQ-WP07-035	Secure communication	A communication mechanism must be available for secure data transfer between two environments of distinct criticalities	Using the example above, data from tire rotation can be classified as safety critical due to a safety feature (ABS/ESP) requiring its data although another feature (tire pressure) with a lower safety impact also requiring it. This adds a need for mixed-criticality systems to supply secure communication between partitioned systems to allow this type of data sharing without losing the highest level of certification.	BN_T7.5_03	ST7.5.2 ST7.5.3 ST7.5.4
HL-REQ-WP07-036	Android	The Android Operating System shall be the runtime environment used as the Infotainment system	As one of the use case objectives, the usage of Commercial Off-the-Shelf software brings the benefit of low user learning curve with an good application base and ease of application development and deployment.	BN_T7.5_04	ST7.5.5
HL-REQ-WP07-037	RTOS	A real time operating system shall be used as the runtime environment for scheduling hard real time tasks.	Safety critical features usually require timming constraints that can only be met with real time scheduling. Therefor, a realtime operating system is required.	BN_T7.5_05	ST7.5.5
HL-REQ-WP07-038	Infotainment	The infotainment system shall use the Android applications for its multimedia purposes	Using Android as a base platform for the infotainment system allows not only for the usage of already available apps and ease of application development to deliver multimedia and navigation features required, but also to have already target hardware platform vendor developed drivers.	BN_T7.5_06	ST7.5.3 ST7.5.4 ST7.5.5
HL-REQ-WP07-039	System devices	HW resources shall be managed by their assigned OS	To keep development effort to a minumum and increase performance, the Android OS must have direct access to hardware and as such it is	BN_T7.5_07	ST7.5.5

			responsible of managing each device it's designed to have access to. For example, the 3G graphics card requires closed source drivers to deliver peak performance that would render certification near impossible. On the other hand, the CAN bus must be tightly controlled due to safety requirements and can only be managed by certified software.		
HL-REQ-WP07-040	eCall system	An eCall system shall be developed as a critical real time task on the RTOS runtime environment	A use case requirement to have an safety critical eCall task that sends a warning whenever a crash would occur. This improves rescue services with information such as location, number of passengers and optionaly their blood type.	BN_T7.5_08	ST7.5.4, ST7.5.5
HL-REQ-WP07-041	Software Containment	The distinct levels of criticality shall be addressed using hardware assisted virtualization.	This is the key part which allows two or more software components to run within the same hardware platform with distinct levels of safety impact and certification. Hardware assisted virtualization greatly improves performance by, for example, adding an extra unpriviledge mode for software to run.	BN_T7.5_09	ST7.5.2, ST7.5.3, ST7.5.4, ST7.5.5
HL-REQ-WP07-042	Resource partitioning	The execution environment shall allow for space and time partitioning.	A major part in a mixed-criticality platform is the ability of secure partition of space and time resources with each software component restricted to a pre-determined time of execution as well to a limited ammount of hardware resources. This requires the platform be able to secure resource access and enforce time constrains as designed, in a way that a task or RTE cannot access outside expected boundries and execute for no longer than expected.	BN_T7.5_10	ST7.5.2, ST7.5.3, ST7.5.4 ST7.5.5
HL-REQ-WP07-043	Resource takeover	The execution environment shall support non-critical resource takeover by	Using the example of the 3G device, in normal usage it can be managed	BN_T7.5_11	ST7.5.2 ST7.5.3 ST7.5.4

		critical component initially determined as non-critical.	by the non-critical RTE and only when a crash occurs, the eCall application requires 3G connectivity to send out information. This means that the device must be secured from the non-critical RTE access, reset and then restarted as a safety critical resource.		ST7.5.5
HL-REQ-WP07-044	Inter-core communication	The execution environment shall support an inter-core communication API and be able to send a signal between cores.	Secure data communication between tasks and/or RTE is crucial to a mixed-criticality platform. Again using the example of ESP and tire pressure systems, the data comes from the same source while the targets differ in criticality, adding the possibility of multi-core, each task can be running concurrently and demanding information that must be available per-task/per-core without requiring lock. Another application is the communication between both critical and non-critical RTE each running in distinct cores with distinct criticalities that also must share information between.	BN_T7.5_12	ST7.5.3 ST7.5.4 ST7.5.5
HL-REQ-WP07-045	Execution monitoring	The execution environment should at least support the monitoring of CPU utilization and idle time.	To evaluate correct space and time partitioning as well software faults, the platform must have a monitoring solution that allows this data to be stored so it can be later analysed.	BN_T7.5_13	ST7.5.5
HL-REQ-WP07-046	Core affinity	The execution environment should allow binding a function to a specific core.	In order to avoid cache misses/flushing and interrupt migration each RTE shall have their cores assigned at boot as defined by the design used. This could, for example, attribute 2 cores for the Android OS and the remaining 2 to the RTOS.	BN_T7.5_14	ST7.5.5

9.4 Evaluation plans

The use case evaluation shall be performed on how the technologies showcased in the demonstrator provide feasible solutions to the high level requirements and associated business needs.

These may be categorized as expectations versus the delivered solutions:

A. **Exceeds expectations**
The solution given goes beyond the expectations;

B. **Totally fulfilled**
The solution completely fulfils the expectations;

C. **Partially fulfilled**
The solution given does bring something new but partial covers the expectations;

D. **Not fulfilled**
The solution although attempted, does not fulfill the expectations;

E. **Not covered**
There was no attempt to implement a solution that covers a specific need.

Table 22: Evaluation plans for use case "Infotainment and eCall Application Multi-Critical Application"

Requirement ID	Short Description	Expectations
HL-REQ-WP07-033	Mixed criticality tasks	It's expected that the platform supports tasks with distinct levels of criticality within the same hadrware platform. Their criticality being staticaly assigned at design level, the platform must ensure proper partitioning and scheduling to meet each task requirement.
HL-REQ-WP07-034	Multicore tasks	It's expect that the platform supports 1. Real-time task scheduling in a multi-core CPU; 2. With each task requirements set at design, the platform must ensure their correct scheduling on defined CPU cores;
HL-REQ-WP07-035	Secure communication	It's expected that the platform supports a communication mechanism that: 1. Enables runtime environments of distinct safety-critical requirements to communicat safely with each other; 2. Supports two-way data communication; 3. Support for safety analysis with fault injection support; 4. Support for online monitoring;
HL-REQ-WP07-036	Android	It's expected that the infotainment platform be the Android Operating System running as a contained non-critical non-trusted RTE.
HL-REQ-WP07-037	RTOS	It's expected that the platform supports fixed-priority mixed-criticality task scheduling on designed CPU cores with support for task pre-emption.
HL-REQ-WP07-038	Infotainment	It's expected that the infotainment system shall use the Android applications for its multimedia purposes
HL-REQ-WP07-039	System devices	It's expected that the platform supports hardware management with: 1. Support for device security access policies; 2. Support for memory management and memory access policies;
HL-REQ-WP07-040	eCall system	It's expected that the platform provides an eCall system as a safety-critical real time task.
HL-REQ-WP07-041	Software Containment	It's expected that the platform supports distinct levels of criticality, using hardware assisted functionality, and use this to contain the infotainment RTE in both time and space partitions from the RTOS runtime environment.
HL-REQ-WP07-042	Resource partitioning	It's expected that the platform supports space and time partitioning for both tasks and RTE of distinct safety-critical requirements.
HL-REQ-WP07-043	Resource takeover	It's expected that the platform supports resource takeover by a safety-critical functionality for specific resources already defined in the design.

HL-REQ-WP07-044	Inter-core communication	It's expected that the platform supports a inter-core communication mechanism that: 1. Enables tasks of distinct safety-critical requirements to communicat safely with each other over distinct CPU cores; 2. Supports two-way data communication; 3. Support for safety analysis with fault injection support; 4. Support for online monitoring;
HL-REQ-WP07-045	Execution monitoring	It's expected that the platform supports online monitoring of some shared resources, such as CPU utilization and idle time.
HL-REQ-WP07-046	Core affinity	It's expected that the platform supports binding a function to a specific core.

10. Appendix 6: T7.6 UC Next Generation Electronic Architecture for Commercial Vehicles

10.1 KPIs

Table 23: KPIs for use case "Next generation electronic architecture for commercial vehicles"

No	Title	Description	BN relation
KPI_T7.6_01	Reduction of number of control units	Reduced total number of ECUs in next generation E/E architecture for commercial vehicles by 20% with the respect to the current-generation E/E architecture, assuming same level of functionalities and features realized.	BN_T7.6_01
KPI_T7.6_02	Definition of architecture evaluation methods	Defined systematic evaluation methods to compare alternative E/E architectures to identify the most suitable one that meets a given set of requirements as well as supports scalability, flexibility, reusability and adaptability. Such evaluation methods should include embedded network communications.	BN_T7.6_02
KPI_T7.6_03	Definition of concepts, methods and tools for partitioning and allocation of software	Defined concepts, methods and tools for efficient partitioning and allocation of functionalities as well as software across the multicore ECUs at the E/E architecture level and across the processor cores within a particular ECU to maximize performance and resource utilization.	BN_T7.6_03
KPI_T7.6_04	Definition of methodologies for dependability enhancement	Defined methodologies and guidelines for utilization of available resources of multicores to improve product quality and safety by enhancing dependability (functional safety, reliability, fault tolerance, security, and availability) both at the E/E architecture level and at the ECU level.	BN_T7.6_04
KPI_T7.6_05	Increase of the degree of integration of applications of mixed criticality levels	Increase integration of functionalities with different criticality levels within a single ECU by 20% with respect to the current level of functionality integration by exploiting the potential of multicores.	BN_T7.6_05
KPI_T7.6_06	Definition of communications strategies	Defined strategies for Inter-ECU (at the E/E architecture level) and intra-ECU communication (network topology, protocol, bandwidth, delay, technology).	BN_T7.6_06
KPI_T7.6_07	Definition of Service-oriented architecture concepts	Defined concepts including viability and applicability of Service-oriented Architecture (SoA) for real-time functional safety-related automotive systems.	BN_T7.6_07
KPI_T7.6_08	Definition of runtime optimization concepts	Defined concepts including viability and applicability of run-time optimizations for real-time safety-related automotive systems.	BN_T7.6_08

KPI_T7.6_09	Definition of concepts and recommendations for AUTOSAR support for multicore, SoA and runtime optimization.	Defined concepts and recommendations to (a) improve multicore support for mixed-critical systems in future releases of AUTOSAR as well as other relevant standards and specifications, and (b) suggest required changes in AUTOSAR to support SoA and run-time optimizations.	BN_T7.6_09
KPI_T7.6_10	Definition of concepts for mixed OS support	Defined concepts for mixed OS support in one and the same ECU.	BN_T7.6_10

10.2 Sub task descriptions

Figure 13 shows the tasks within this UC and the relation of each task with other tasks within this UC as well as with other technical WPs. While the subtasks ST7.6.2 – ST7.6.5 are not independent of each other the necessary interactions across the tasks is ensured in ST7.6.1. The scope and more specific description of the use case are defined within this task.

Requirements are derived and collected in ST7.6.1 along with definitions of use case specifications, evaluation criteria and evaluation methodology. The requirements are communicated to the relevant tasks of this UC, Automotive Living Lab (WP7), and the other technical WPs (WP1 SP_SoA – Embedded Systems Architecture, WP2 SP_Application Programmability and Static/Offline System Software, WP3 SP_Dynamic Runtime Environments and Services, WP4 SP_Multi-Core Hardware Architectures and Concepts, WP5 SP_System Design Platform, Tools, Models and Interoperability, WP6 SP_System Qualification and Certification) to perform the required technical activities.

General concepts, techniques and tools are developed in the respective technical WPs. Concepts, methods and tools developed in other WPs are to be customized to meet the requirements and specifications of the use case.

Techniques and tools that are developed and available in other industrial domains, for example, avionics are to be considered as well. On the other hand, techniques specific to the use case is investigated and developed within each task of this UC. At the end, customization and integration is made to perform demonstration and evaluation of the use case.

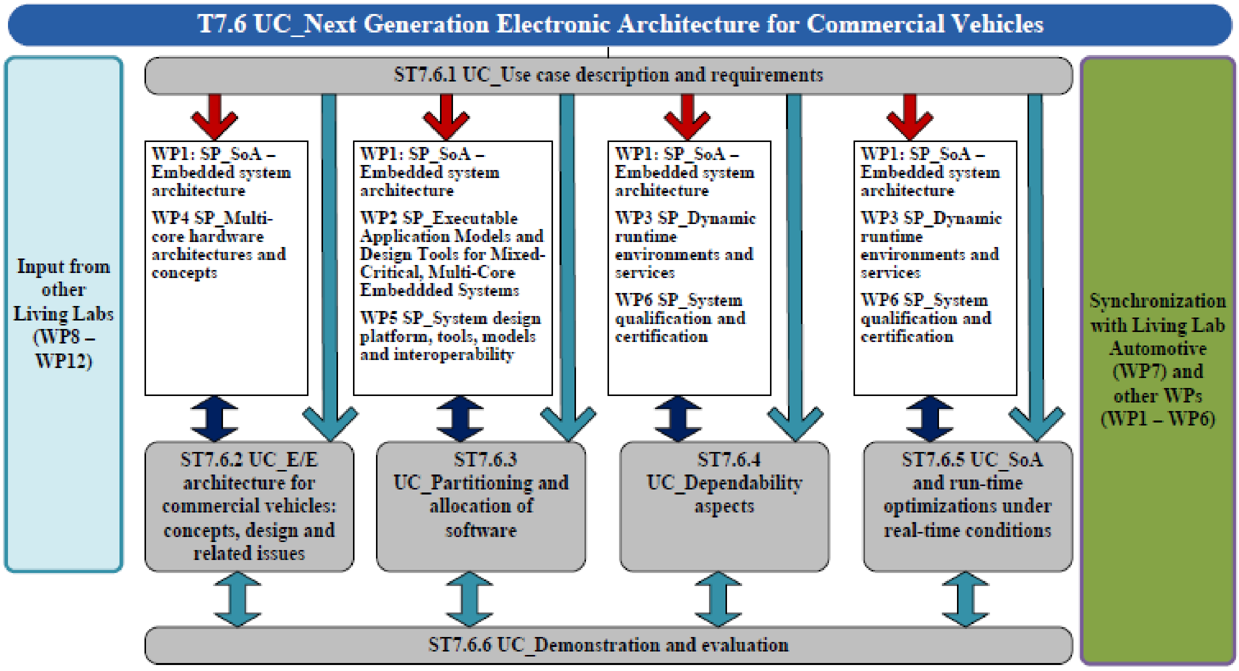

T7.6 UC_Next Generation Electronic Architecture for Commercial Vehicles
ST7.6.1 UC_Use case description and requirements
Input from other Living Labs (WP8 – WP12)
WP1: SP_SoA – Embedded system architecture
WP4 SP_Multi-core hardware architectures and concepts
WP1: SP_SoA – Embedded system architecture
WP2 SP_Executable Application Models and Design Tools for Mixed-Critical, Multi-Core Embeddded Systems
WP5 SP_System design platform, tools, models and interoperability
WP1: SP_SoA – Embedded system architecture
WP3 SP_Dynamic runtime environments and services
WP6 SP_System qualification and certification
WP1: SP_SoA – Embedded system architecture
WP3 SP_Dynamic runtime environments and services
WP6 SP_System qualification and certification
Synchronization with Living Lab Automotive (WP7) and other WPs (WP1 – WP6)
ST7.6.2 UC_E/E architecture for commercial vehicles: concepts, design and related issues
ST7.6.3 UC_Partitioning and allocation of software
ST7.6.4 UC_Dependability aspects
ST7.6.5 UC_SoA and run-time optimizations under real-time conditions
ST7.6.6 UC_Demonstration and evaluation

Figure 13: Subtasks of T7.6 and interaction with WP7 and technical WPs (1-6).

10.2.1 ST7.6.1 Use case description and requirements

The activities performed within this task are as follows:

- Identification of requirements specific to the use case based on the already identified business needs of the automotive industry from commercial vehicles viewpoint. Requirements will cover the following areas of interests:
  - Logical architecture and physical E/E architecture
  - Mapping between logical and physical architectures, partitioning and allocation of functionalities across the ECUs
  - Partitioning and allocation of software within a single ECU and across a set of ECUs specific to engine controllers
  - Functional requirements, for example, performance
  - Non-functional requirements, for example, framework for dependability and security
  - Communication issues (inter-ECU and intra-ECU)
  - Separation, i.e., freedom from interferences, among applications with different criticalities and management of shared resources
  - Support for scalability, flexibility, adaptability and reusability
  - SoA and run-time optimizations for real-time safety-related systems
- Specific description of the use case, and definition of plans and means of demonstration.
- Identification of criteria and metrics to perform systematic evaluation and selection of an E/E architecture.This includes definition of a baseline E/E architecture that can be used to estimate cost reduction, performance improvement, complexity reduction, dependability improvement and so on.
- Identification of methodology to perform evaluation of concepts, techniques and tools.

Relation to deliverables: This task contributes to common WP7 deliverable (D7.1), common T7.6 deliverable (D7.13) and an internal report specific to T7.6.

Relation to business needs: BN_WP7_T7.6_01 - BN_WP7_T7.6_09

Relation to WPs: WP1 – WP6 and Living Labs of other industrial domains

10.2.2 ST7.6.2 E/E architecture for commercial vehicles

The activities to be performed within this task are as follows:

- Concepts and issues related to logical and physical E/E architecture. This includes activities towards Integrated Vehicular Architecture similar to the concepts of IMA in the avionics industry.
- Concepts and techniques to reduce the total number of ECUs and the variants across the ECUs with respect to the current-generation E/E architecture for commercial vehicles.
- Application of criteria and metrics to evaluate functional requirements such as performance of the E/E architecture. This includes multi-criteria optimization as well as supports for scalability, flexibility, reusability and adaptability.
- Communication issues at the E/E architecture level.
- Partitioning and allocation of functionalities across the ECUs. Also, this will provide input to ST7.6.3 to perform partitioning and allocation of software on multicore ECUs.
- Impact of processor architectures on the E/E architecture and processor hardware support for mixedcriticality applications.

Relation to deliverables: This task contributes to the common WP7 deliverable (D7.2), common T7.6 deliverables (D7.13 and D7.14) and an internal report related to the E/E architecture.

Relation to business needs: BN_WP7_T7.6_01, BN_WP7_T7.6_02, BN_WP7_T7.6_03, BN_WP7_T7.6_05, BN_WP7_T7.6_06, BN_WP7_T7.6_08, BN_WP7_T7.6_09

Relation to WPs: WP1, WP4 and Living Labs of other industrial domains

10.2.3 ST7.6.3 Partitioning and allocation of software

The activities that will be performed within this task are as follows:

- Partitioning and allocation of software across the ECUs. This includes system-level timing analysis as well as concepts and methods to maximize performance and resource utilization.
- Partitioning and allocation of software across the processor cores within a single multicore ECU. This includes concepts, methods and tools related to worst-case execution time (WCET) analysis, scheduling, load balancing and synchronization.
- Separation, i.e., freedom from interference, among applications with different criticalities and management of shared resources.
- High integration of functionalities and improvement of existing functionalities.
- Tailor methodologies and tools developed in WP2 - T2.6 *"Offline methods and tools supporting the needs of the automotive domain"* to the special requirements of this use case in order to achieve the related business goals. One main goal is the consideration of existing standards and defacto standards (ISO 26262 for functional safety and AUTOSAR for software components)

Relation to deliverables: This task will contribute to the common WP7 deliverable (D7.2), common T7.6 deliverables (D7.13 and D7.14), and an internal report related to the partitioning and allocation of software across the ECUs and across the processor cores within an ECU.

Relation to business needs: BN_WP7_T7.6_01, BN_WP7_T7.6_03, BN_WP7_T7.6_05, BN_WP7_T7.6_06,
BN_WP7_T7.6_08, BN_WP7_T7.6_09

Relation to WPs: WP1, WP2, WP5 and Living Labs of other industrial domains

10.2.4 ST7.6.4 Dependability aspects

The activities that will be performed within this task are as follows:

- Support for dependability and security related aspects in terms of flexibility, scalability adaptability and reconfigurability at the E/E architecture level and single ECU level.
- Technical safety concept to meet functional safety requirements according to ISO 26262 in the context of multicores for "fail-stop" and "fail-operational" E/E systems.
- Reliability, availability and fault tolerance as part of AUTOSAR platform development to integrate robustness concept within AUTOSAR for multicores.
- Security support in the AUTOSAR platform for attack detection and protection.
- Identify concepts, methods, tools and tool chains to achieve dependability and security goals within this task. Adapt concepts, methods, tools and tool chains from other technical WPs (WP1, WP5, WP6).

Relation to deliverables: This task will contribute to the common WP7 deliverable (D7.2), common T7.6 deliverables (D7.13 and D7.14), an internal report related to the dependability aspects, and an implementation of automotive dependability and security framework component.

Relation to business needs: BN_WP7_T7.6_04, BN_WP7_T7.6_09

Relation to WPs: WP1, WP3, WP6 and Living Labs of other industrial domains

10.2.5 ST7.6.5 SoA and run-time optimizations under real-time conditions

This task investigates the suitability of SoA and run-time optimizations for real-time safety-related automotive systems along with identification of associated challenges and implications.

The SoA shall enable a NASREM [Albus1996] type of layered control hierarchy to be a basic structure of the application's architecture. It is highly desired if the complexity of a service can be encapsulated to provide a consistent interface to the requesting application. For example, the fact that a service implementation is distributed over multiple nodes and cores, local or remote (location transparency), should be hidden (access transparency). Allocation and partitioning will be done in collaboration with ST7.6.3. The quality of the service, e.g. in terms of measurable end-to-end latency, is an abstraction to replace implementation details. Methods and tools from WP3 and WP5 will be explored to achieve the aims of this work task.

The SoA applied here follows closely the design pattern proposed in WP1, e.g the common definition of metadata. For the investigation of applicability, a set of system wide services, including but not limited to the ones enumerated below, is to be defined.

- Computationally intense run-time optimization could be requested at any level in the control hierarchy and would be suitable to implement as an autonomous service on top of the set of operating systems and communication protocols used. The service must be provided to multiple simultaneous requests, not necessarily arriving periodically. The possibility to parallelize the optimization algorithm and partition the execution on available cores will be explored, including dynamic adaptation.
- A repository can be implemented as a database or as a file system. The physical non-volatile storage can be distributed and its size can be dynamically scalable. An interface needs to be designed to allow a uniform access sequence to the repository service. Typical data is internal diagnostic messages and binary images.
- External network communication is a system wide service primarily requested by the upper levels of the control hierarchy. The V2X communication allows for advanced traffic related features to increase safety. A typical communication device is a wireless access point. A network service has to offer e.g. encryption for security and session control to detect omitted messages.

- A human user interface service, is a system wide functionality necessary also for embedded distributed computer systems. This could be implemented as a terminal server where a technician with proper credentials can log in and request information or the execution of administration services. Typical applications are workshop maintenance to perform software upgrade and error messages presented to the driver on the cluster common resource display.
- A graceful degradation application, using a TTEthernet based, time-triggered data communication approach will provide safety-relevant functionality. This contribution will demonstrate dynamic reconfiguration during run-time in order to handle ECU failures in safety-relevant applications. Links to WP1 (architecture), WP3 (dynamic reconfiguration tools) and WP4 (TTEthernet based hardware platform) will be integrated in this part.

Relation to deliverables: This task contributes to the common WP7 deliverable (D7.2), common T7.6 deliverables (D7.13 and D7.14), and one internal report related to SoA and run-time optimizations for real-time safety-related automotive systems. The internal report and common part of the deliverable will provide suggestions on how AUTOSAR needs to be evolved to support SoA and run-time optimizations.

Relation to business needs: BN_WP7_T7.6_07, BN_WP7_T7.6_08

Relation to WPs: WP1, WP3, WP6 and Living Labs of other industrial domains

10.2.6 ST7.6.6 Demonstration and evaluation

The activities performed within this task are as follows:

- Integration of techniques and tools developed within this use case to facilitate demonstration and evaluation of the use case.
- Integration of techniques and tools that are developed in other WPs and are customized for the use case to facilitate demonstration and evaluation of the use case.
- Demonstration of the use case based on the requirements, specifications, and evaluation criteria, metrics and methods as defined in ST7.6.1.
- Evaluation of the use case to identify whether the achieved results are in line with the expected results in relation to use case requirements and project requirements.
- Provide input and suggestions to the future releases of the relevant standards and specifications (e.g., ISO26262, AUTOSAR).

Relation to deliverables: This task will contribute to the common WP7 deliverable (D7.2), common T7.6 deliverable (D7.14), and one internal report regarding use case demonstration and evaluation results.

Relation to business needs: BN_WP7_T7.6_01 - BN_WP7_T7.6_09

10.3 High level requirements

The high level requirements derived from the use case are shown in the table below. (The numbering of the requirements is consistent with the numbering at project level.)

Table 24: High level requirements for use case “Next generation electronic architecture for commercial vehicles”

Requirement ID	Short description	Description	Rationale	BN relation	Sub-task relation
HL-REQ-WP07-023	Generic computational ECUs.	Technology for generic configurable ECUs with high computational capabilities shall be	Traditionally new ECUs are added when new functionality is introduced and as a	BN_T7.6_01	ST7.6.2, ST7.6.3

Requirement ID	Short description	Description	Rationale	BN relation	Sub-task relation
		developed and documented.	consequence many different ECU variants exists. High capacity generic ECUs carrying multiple applications and possible to easily configure for different kinds of applications would be of benefit in this regard.		
HL-REQ-WP07-024	Architecture evaluation methods.	Architecture evaluation methods shall be defined to be used for comparing alternative E/E architectures, including embedded network communications, to identify the most suitable one that meets a given set of requirements as well as supports scalability, flexibility, reusability and adaptability.	To be able to evaluate architecture alternatives systematic evaluation methods is needed in order to support the selection of alternatives.	BN_T7.6_02	ST7.6.2, ST7.6.5
HL-REQ-WP07-025	Software partitioning and allocation.	Methods and tools for partitioning and allocation of software across the multicore ECUs at the E/E architecture level and across the processor cores within a particular ECU in order to maximize performance and resource utilization shall be developed.	With multicore ECUs there is a need for methods for partitioning and allocation of base SW as well as application SW across EUCs and across cores in an ECU. Such methods should be supported by tools as an integrated part of the development environment.	BN_T7.6_03	ST7.6.3
HL-REQ-WP07-026	Dependability enhancement.	Concepts for dependability (functional safety, reliability, fault tolerance, security, and availability), both at the E/E architecture level and at the ECU level, shall be defined in the context of available resources of multicores.	Enhancement of dependability in the multicore context in order to ensure and improve product quality and safety.	BN_T7.6_04	ST7.6.4
HL-REQ-WP07-027	Mixed criticality support.	Separation, i.e. freedom from interferences, among	The integrity of applications when executing different	BN_T7.6_05	ST7.6.3, ST7.6.4

Requirement ID	Short description	Description	Rationale	BN relation	Sub-task relation
		applications with different criticality levels (as defined in ISO 26262) shall be ensured in one and the same ECU.	applications at different cores these applications must be guaranteed.		
HL-REQ-WP07-028	Communications guidelines.	Guidelines for Inter-ECU at the E/E architecture level and intra-ECU communication (including topics network topology, protocol, bandwidth, delay, technology) in multicore based architectures shall be defined.	With new architecture concepts based on multicore ECUs, new demands on the communications system can be foreseen. E.g. highly critical application communication mixed with less critical communication may raise the needs of dynamically allocated high priority channels for event driven data on the account of low priority data.	BN_T7.6_06	ST7.6.2
HL-REQ-WP07-029	Service-oriented architecture concepts.	Concepts and techniques related to Service-oriented Architecture (SoA) for real-time functional safety-related automotive systems shall be defined.	SOA for embedded systems is a new interesting approach that needs to be investigated in the light of new architecture concepts with central computational nodes and light-weight I/O nodes in order to make implementation af applications more independent of underlaying platform.	BN_T7.6_07	ST7.6.5
HL-REQ-WP07-030	Runtime optimization.	Concepts and techniques related to run-time optimizations for real-time safety-related automotive systems shall be defined.	Multicore support with high performance capabilities gives the possibility to have faster and more precise control algorithms that can be adapted and optimized during run-time.	BN_T7.6_08	ST7.6.5
HL-REQ-WP07-031	AUTOSAR support for multicore, SoA and runtime optimization.	Recommendations shall be formulated to (a) improve multicore support for mixed-critical systems in future releases of AUTOSAR as well as	Any implications of SOA and run-time optimization on the AUTOSAR standard can be presented to the AUTOSAR consortium for	BN_T7.6_09	ST7.6.2, ST7.6.3, ST7.6.4, ST7.6.5

Requirement ID	Short description	Description	Rationale	BN relation	Sub-task relation
		other relevant standards and specifications, and (b) suggest required changes in AUTOSAR to support SoA and run-time optimizations.	possible standard modifications.		
HL-REQ-WP07-032	Mixed OS support.	Mixed OS, i.e. AUTOSAR OS and RT Linux, shall be supported in one and the same ECU.	Autosar is needed in order to communicate on the vehicle bus. Autosar is however not enough to run many of the new applications with demands on different communication stacks, local databases and other functionality that is part of a complex (RT)OS.	BN_T7.6_10	ST7.6.2, ST7.6.3, ST7.6.4

10.4 Evaluation plans

This section presents the preliminary evaluation plan for the high level requirements that have already been identified for the use case “Next generation electronic architecture for commercial vehicles” (T7.6). The evaluation plan will be refined in future.

Table 25: Evaluation plans for the high level requirements of the use case “Next generation electronic architecture for commercial vehicles”

Requirement ID	Short description	Evaluation plans
HL-REQ-WP07-023	Generic computational ECUs.	The objective is to investigate the feasibility of powerful computational nodes and generic I/O nodes as the building blocks of next generation E/E architecture. ▪ Investigate how computation and I/O can be separated so that E/E architectures can be based on more powerful and centralized computational nodes alongside generic and light-weight I/O nodes. ▪ Investigate the potential impact of E/E architectures based on computational node and generic I/O node on system design and development. ▪ Propose several alternative physical E/E architectures to evaluate the suitability of such architecture to meet the requirements.
HL-REQ-WP07-024	Architecture evaluation methods.	The objective is to systematically evaluate E/E system architecture candidates to find an architecture that meets a given set of requirements. ▪ Identify and define a set of metrics to evaluate (both qualitatively and quantitatively) alternative E/E architectures. ▪ Define a set of alternative E/E architectures that will be evaluated using the defined metrics. ▪ Develop method and tool support to apply the metrics on the E/E architectures under evaluation. ▪ Apply the method by using the tool support to evaluate the alternative architectures based on the defined metrics to find the most suitable one.

Requirement ID	Short description	Evaluation plans
HL-REQ-WP07-025	Software partitioning and allocation.	The objectives are to perform (a) mapping of logical architecture to physical architecture (architecture-level), and (b) partitioning and mapping of software across multiple cores of a multicore ECU (single ECU-level). ▪ Identify state-of-the-art and state-of-the-practice ▪ Exploit graph partitioning algorithms, methods and tools; adapt and customize the algorithms, methods and tools. ▪ Develop method and tool support to visualize functionalities and software at different abstraction levels to support software design for multicores ▪ Define objective and cost functions to perform architecture- and ECU-level partitioning and mapping. ▪ Investigate and apply method and tool support for timing analysis. ▪ Investigate AADL for automotive E/E systems to model software and hardware architectures. ▪ Investigate and apply method and tool support for scheduling.
HL-REQ-WP07-026	Dependability enhancement.	The objective is to exploit the potential of multicores to increase dependability (functional safety, fault tolerance, availability, reliability) at the single ECU-level and architecture-level. ▪ Investigate safety mechanisms in the context of ISO 26262 and AUTOSAR. ▪ Implement a set of safety mechanisms in line with AUTOSAR. ▪ Evaluate the effectiveness of the implemented mechanisms in fulfilling the functional safety requirements as per ISO 26262. ▪ Evaluate the resource efficiency (e.g., impact on CPU load, timing and memory footprint) of the mechanisms.
HL-REQ-WP07-027	Mixed criticality support.	The objective is to develop solutions for preventing interferences among partitions (i.e., ensure freedom from interferences among applications with different safety criticality levels – e.g., ASIL as per ISO 26262). ▪ Investigate state-of-the-art and state-of-the-practice. ▪ Identify the detailed requirements and use cases. ▪ Adapt method and tool to support designing systems with different safety criticality levels.
HL-REQ-WP07-028	Communications guidelines.	The objective is to provide guidelines to meet the communication requirements of the next generation E/E architectures. ▪ Network topology. ▪ Protocol and communication technologies. ▪ Freedom from interferences with respect to communication.
HL-REQ-WP07-029	Service-oriented architecture concepts.	The objective is to investigate the feasibility of SoA for automotive E/E systems. ▪ Investigate principles and design issues of SoA-based logical E/E architecture, ▪ Identify technical, functional and non-functional requirements for SoA-based architecture for the automotive systems. ▪ Investigate the impact of SoA-based approach on process, method and tool for system as well as software design, implementation and verification. ▪ Investigate possible realization/implementation (prrof-of-concept implementation) of SoA-based physical E/E architecture.
HL-REQ-WP07-030	Runtime optimization.	▪ Investigate the feasibility of implementing run-time monitoring mechanisms to increase error detection capabilities of the E/E system by exploiting the additional computational power of the next-generation multicore ECUs. ▪ Investigate mechanisms (possibility and feasibility) to support re-configuration at runtime to improve error handling capabilities of the E/E systems. ▪ Investigate the possibility to have faster and more precise control algorithms that can be adapted and optimized during run-time.

Requirement ID	Short description	Evaluation plans
HL-REQ-WP07-031	AUTOSAR support for multicore, SoA and runtime optimization.	▪ The current support for multicores, SoA and run-time optimization in AUTOSAR will be studied to identify the state-of-the-practice. ▪ Based on the results of evaluation of other high level requirements (HL-REQ-WP07-023 – HL-REQ-WP07-030), recommendations will be provided to the relevant AUTOSAR working groups for future revisions of the relevant concepts and speficifications.
HL-REQ-WP07-032	Mixed OS support.	The objective is to support multiple OSs, including AUTOSAR RTOS on the same hardware platform where the time critical parts/applications can be started first. ▪ Investigate the feasibility of running another OS (e.g., Linux) alongside AUTOSAR OS on the same hardware platform so that the AUTOSAR OS starts within the required time. ▪ Investigate and develop solutions for efficient (e.g., low-latency) communication. ▪ Investigate and develop solutions for load balancing.