



Distributed REal-time Architecture for Mixed criticality Systems

Final Community Building Report D 9.1.3

Project Acronym	DREAMS	Grant Agreement Number	FP7-ICT-2013.3.4-610640	
Document Version	1.0	Date	2017-09-30	Deliverable No. 9.1.3
Contact Person	Kevin Chappuis	Organisation	VOSYS	
Phone	+33 6 63 25 48 52	E-Mail	k.chappuis@virtualopensys-tems.com	

Contributors

Name	Partner
Kevin Chappuis	VOSYS
Simon Barner	FORTISS
Miltos Grammatikakis	TEI
Maria Cristina Zubia	IKL
Imanol Martinez	IKL
Marcello Coppola	ST
Hamidreza Ahmadian	USIEGEN
Mohammed Abuteir	TTT

Table of Contents

Contributors	2
1 Introduction	4
1.1 Position of the Deliverable in the Project	4
1.2 Contents of the Deliverable	4
2 Goals of Community Building.....	5
3 Community Building Platform.....	6
3.1 Mixed-Criticality Forum	6
3.2 Code repository updates.....	8
3.2.1 IKL open-source: Cross-domain mixed-criticality patterns	9
3.2.2 VOSYS open-source: Scheduling heuristics and coordination for KVM	12
3.2.3 TEI open-source: Soft real-time ECG, Mem/NetGuardXt, on-chip security.....	14
3.2.4 USIEGEN open-source: DREAMS MCEL.....	15
4 Events.....	16
4.1 Gilching, 2015 event	16
4.2 Prague, 2016 event	17
4.3 Dresden, 2016 event.....	17
4.4 Grenoble, 2016 event	18
4.5 Barcelona, 2016 event	18
4.6 Stockholm, 2017 event	18
4.7 Dubrovnik, 2017 event.....	19
4.8 DREAMS Final Workshop	19
5 Conclusions and Next Steps	21

1 Introduction

This document is the deliverable D9.1.3 of the DREAMS project. It is the last deliverable of task *T9.1 – Community Building* of work package *WP9 - Community building and standardization*. This deliverable *D9.1.3 – Final Community Building Report* presents the activities that have taken place between M24 – M48, with respect to community building, i.e. the dissemination of open-source contributions through a common community building platform, as well as planned events and meetings with the goal of strengthening the collaboration of the Mixed-Criticality forum partners.

1.1 Position of the Deliverable in the Project

The goal of work package WP9 is to steer and increase European research and technology awareness in the area of distributed mixed-criticality and embedded computing systems. Work package WP9 comprises of three tasks: T9.1, T9.2 and T9.3. Task *T9.1 – Community building* aims at building a sustainable community focusing on the results of the DREAMS project and other projects on mixed-criticality systems. Task *T9.2 – Standardization support* aims to provide support towards all standardization efforts emerging from all activities and results of the DREAMS project. Task *T9.3 - Innovation roadmap* aims to help align the academic and industrial research by developing a research and innovation roadmap on the topic of mixed criticality to achieve critical mass and facilitate breakthrough innovations in the medium and long- term.

The deliverable D9.1.3 relates to task T9.1, which sums the following deliverables:

- *Deliverable D9.1.1 Community Repository*: This deliverable provides an overview of the activities related to the start of the community building activities, i.e. the setup of the respective infrastructure for providing services to the community.
- *Deliverable D9.1.2 Intermediate Community Building Report*: This deliverable focuses on the intermediate report on building up a sustainable community for mixed-criticality systems and particularly the “Mixed-Criticality Cluster” (representing the DREAMS, PROXIMA and CONTREX projects).
- *Deliverable D9.1.3 Final Community Building Report*: This deliverable is the final report on the results of the DREAMS project with respect to community building both within the Community Repository and within already existing communities (e.g. HIPEAC).

D9.1.1 and D9.1.2 have been respectively delivered on M18 and M24, therefore, this deliverable (D9.1.3), finalizes the list of Community building activities and results. The confidentiality level of this deliverable is public (PU) and it will be published on the DREAMS website, once approved by European Commission.

1.2 Contents of the Deliverable

In chapter 2, we provide the continuous goals and objectives of Community Building, while updates regarding the Community building platform are described in chapter 3. Chapter 4 lists previous events and venues. Finally, in chapter 5 the conclusions and next steps are presented.

2 Goals of Community Building

As defined in D9.1.1 – Community Repository, the overall goals for the Community Building activities in DREAMS are:

- **Cross-fertilization of the European industry** and strong cooperation between European universities, R&D centers, large enterprises and SMEs. Eased access to leading edge technology, contacts to complementary SMEs, support on the take-up and use of technology and tools as well as support and participation in standardization activities. This is facilitated by *harmonization of mixed-criticality efforts* allowing for long-term reduction of development efforts and increased innovation uptake, by building on the expertise in the European mixed-criticality community and promoting knowledge exchange.
- **Streamlining of research efforts** is achieved by *identifying industrial needs and research challenges in the innovation and research roadmap* that is shared among European stakeholders through the mixed-criticality community building activities.
- **Standardization and coordinated future development** is made possible by a thriving mixed-criticality community that can promote standard practices, i.e. de facto industry standards. Standardization reduces fragmentation and supports further collaboration in European industry, reducing costs and entry-barriers of proprietary solutions.
- **Reinforced competitiveness of European technology** suppliers across the computing spectrum by supporting R&D efforts and support the rapid take-up of project results beyond the project consortium, providing a single ‘hub’ for European R&D on mixed-criticality systems. This comes directly at the advantages particularly of small and medium enterprises that profit from the network and community ranging from component developers to application developers and system integrators.

The second set of activities under the Community Building flag deal with updating and enhancing the current infrastructure set by D9.1.1, which aims to **“Support the mixed-criticality community by facilitating active exchange of ideas as well as technological building blocks”**. In a similar fashion with the goals, the main objectives of Community Building are:

- **Support the Innovation Roadmap:** By developing a roadmap for research and innovation on mixed critically to establish the state-of-the-art in the area and identify research challenges by harnessing the collaborative efforts of the partners.
- **Provide News on Mixed-Criticality Activities:** Through the MCS forum more details are provided with the news section and member activities are highlighted.
- **Mixed-Criticality Projects Visibility:** The MCS community infrastructure provides additional information about the projects in the Mixed-Criticality Cluster and the on-going research activities and status.
- **Catalogue of Project results:** As mentioned in chapter 3, with the updates of the MCS forum website, specific details can be found for the projects’ building blocks and results. In particular the focus is on Meta-models for application and platform modeling, Virtualization components, Simulation environment, Tool support, and Documentation and training material.
- **Code repositories and continuous integration:** On top of the MSC forum website another layer of infrastructure is offered, with the aim of sharing information and software, as well as providing features for versioning control and automated testing. The details for the code repository infrastructure and CI features are mentioned in chapter 3 and 4.

3 Community Building Platform

In this chapter, we list the effort and results on creating a relevant and feature-full Community Building Platform, including the Mixed-Criticality Forum website as well as the DREAMS Code repositories.

3.1 Mixed-Criticality Forum

The Mixed Criticality Forum (MCF) connects DREAMS with other projects in the area of mixed-criticality systems in order to provide a common place for these projects to share news, events, results and general information about mixed-criticality systems (C.F. **Error! Reference source not found.**). In the previous deliverables, D9.1.1 and D9.1.2, the MCF objectives and its development has been described as well as the design, implementation and launch of a common community building platform.

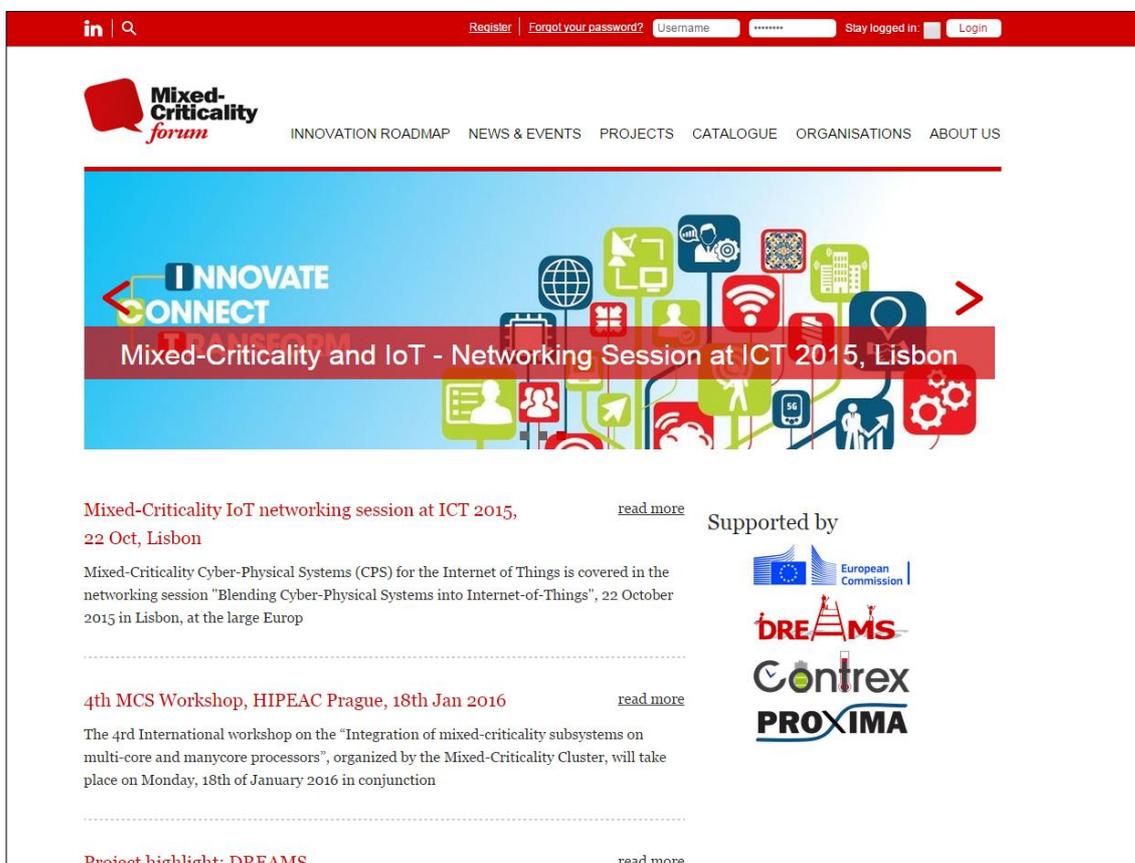


Figure 1: Mixed-Criticality Forum Homepage

During M24 through M48, the community of the MCF is growing. Now, over 85 organizations have registered themselves currently representing 15 European projects as shown in Figure 2 , and with good hope that the platform will be filled with more projects and results in the area of mixed-criticality. Furthermore, the MFC announced in the webpage the mixed-criticality events, such as HIPEAC, workshops and conferences in the area of MCS, etc.



Figure 2: The involved European projects in MCF

Based on the input from partners, the projects results have been added to MCF platforms in the project directly and these results that come from the project directly are highlighted on the MCF home, as depicted in Figure 3.

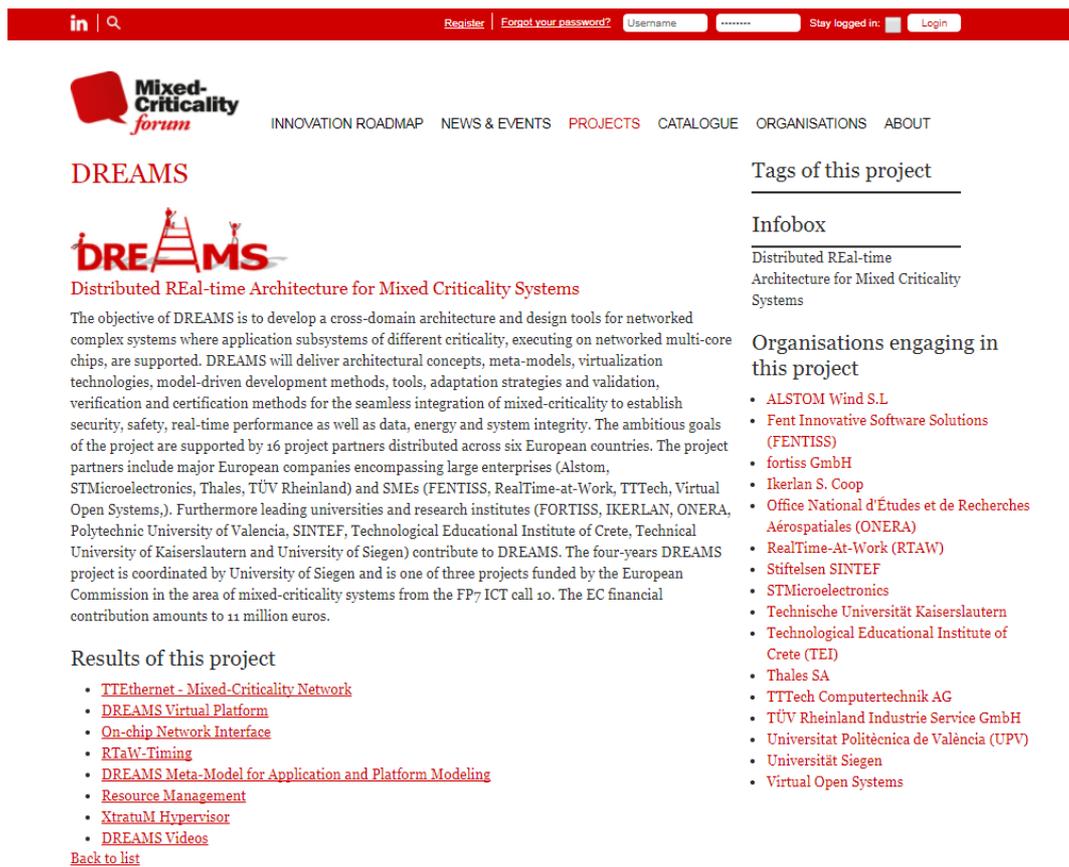


Figure 3: Project Results

The main activities under the Community Building flag in this period was to deal with updating and enhancing the MFC platform to Support the mixed-criticality community by facilitating active exchange of ideas as well as technological building blocks.

3.2 Code repository updates

Part of the community building infrastructure activity is also dedicated to creating a full featured version control and project management system for all users of the Mixed-Criticality Forum. In the previous deliverable, D9.1.1 and D9.1.2, the infrastructure was documented and described as well as the advanced features, such as the continuous integration, which provide methods to immediately test a project's source code by building, testing and deploying on each change of the code base.

During M24 through M48, the GitLab repository infrastructure has been used by many partners for various content, including deliverables, firmware and kernel source code, as well as user-space application code. Indeed, the DREAMS GitLab repository is composed by more than 50 users which are working on 34 projects as shown on Figure 1.

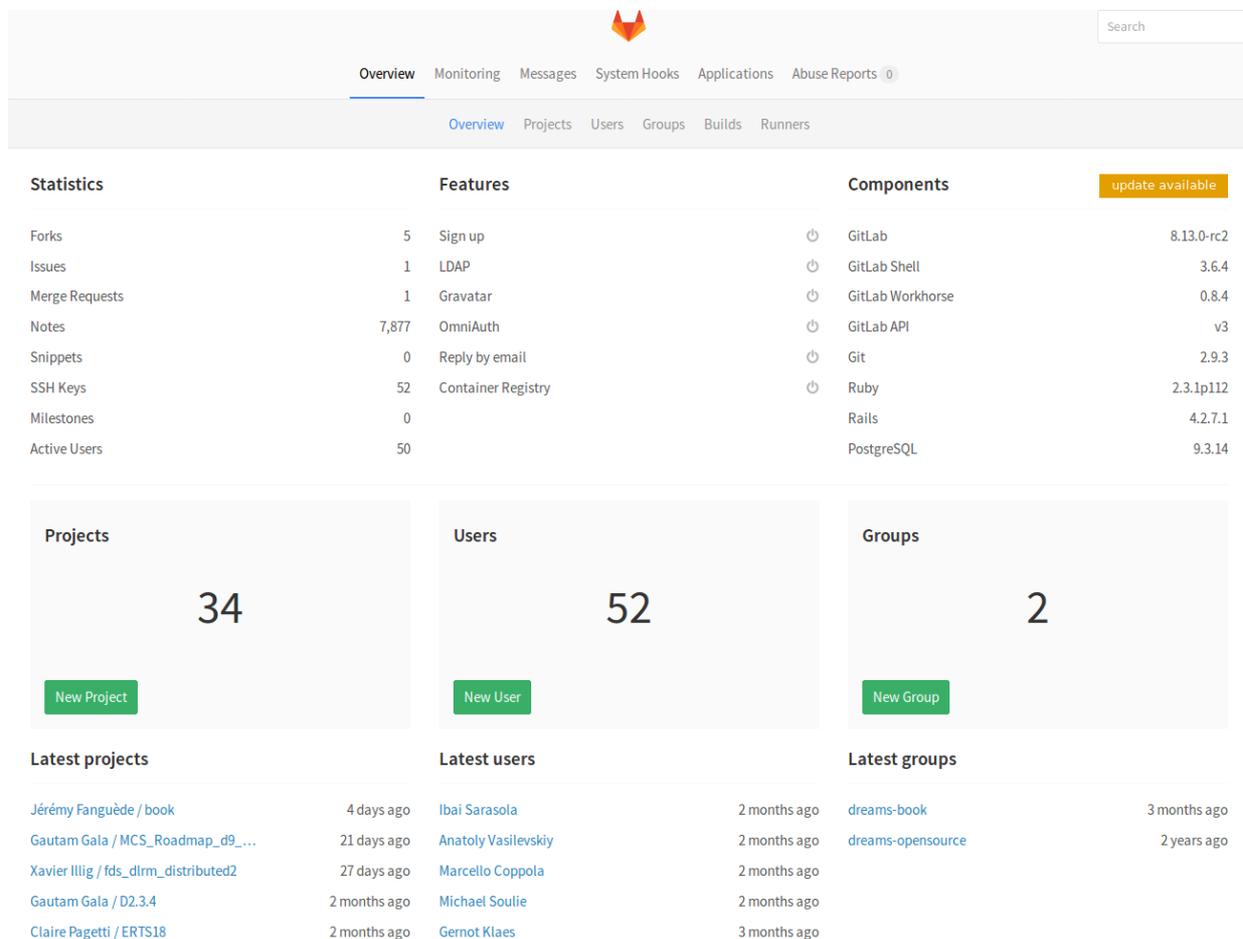


Figure 4: DREAMS GitLab infrastructure overview

As a matter of fact, Figure 2 shows that around 30 project members have actively contributed to the DREAMS book by using the GitLab repository infrastructure, which enables coordinating work on files among multiple users.

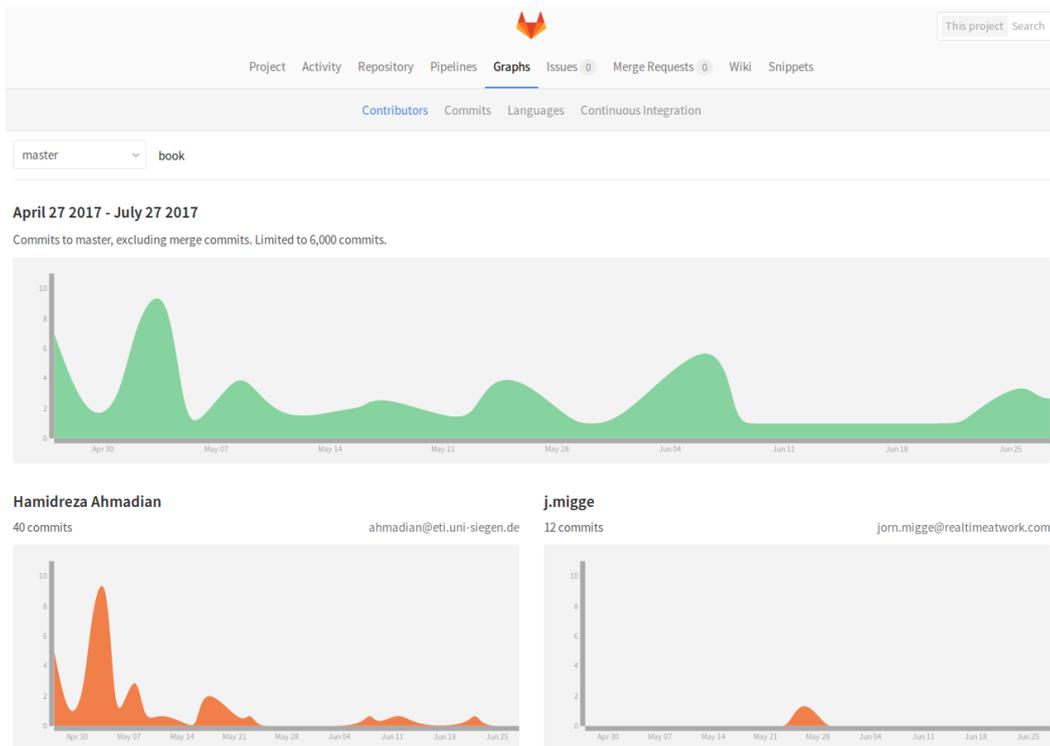


Figure 5: DREAMS book contribution statistics

Additionally, GitLab has been updated from version 7.13.5 to 8.13.0, introducing new features and functionalities, such as new permissions of public/uploads directory to be more restrictive, update configuration management of storage directories, add support for using NFS, etc.

Last but not least, a new group has been created in order to share with the community the project results developed in the context of DREAMS. In this context, the group “dreams-opensource” has been configured as public in order to enable the access of the repository content to any external parties of the DREAMS GitLab repository. In the following sub-sections, the different open-source contributions available on the DREAMS public repository are described.

3.2.1 IKL open-source: Cross-domain mixed-criticality patterns

One of the main results of WP5 has been the definition and implementation of cross-domain mixed-criticality patterns. These patterns aim to guide and support engineers towards solutions that solve commonly occurring problems in the development of mixed-criticality products, from design to verification and validation. The cross-domain patterns which are defined in this deliverable have been identified as the result of the analysis of the IEC 61508 safety-related standard and its application for today’s mixed-criticality systems. The cross-domain patterns have been made available to the community through the DREAMS public repository as follow.

3.2.1.1 Hypervisor related patterns

Critical partition diagnosis. PAT-CPD-00

Partitioned mixed-criticality architecture limits the impact of changes, provides reusability of their parts and reduces the complexity of the system. Partitions can be designed, developed and certified individually with different levels of criticality (e.g., SIL1 – SIL4 according to IEC 61508). If a partition contains a safety critical function that it is considered critical by the system, it should be protected against interferences (temporal and spatial) caused by other partitions.

When dealing with partitioned systems with different criticality, failures caused by the interchange of information are quite probable. The lower criticality functionalities can lead to interferences on the higher criticality functionalities. Therefore, it must be guaranteed that partitions with different criticality level do not influence each other. Two possible sources of interferences can be considered:

1. Temporal interferences generated by multiple accesses in parallel to the shared memory (e.g., by the cores of a multi-core device). The concurrent accesses will compete for accessing to the shared memory cache, which will lead to interferences in temporal domain.
2. Failures in the spatial isolation provided by the hypervisor. It is also found in mono-core architectures.

This pattern aims to provide a generic diagnosis pattern to detect interferences on critical partitions. It provides a scalable set of measures and diagnosis techniques to detect and control failures of critical partitions and guarantee the system's temporal and spatial independences.

Digital I/O Server. PAT-DIOS-00

Digital I/Os are widely used among different system architectures for communication purposes. They can be managed by partitions with different criticality (e.g., safety, non-safety and real-time) but not at the same time. The simultaneous access to a digital I/O by more than one partition usually leads to an error that jeopardizes the system.

The proposed solution within this pattern is based on the implementation of a Digital I/O server partition, which manages the digital I/Os of the mixed-criticality system. The Digital I/O Server (DIOS) is a consistent concurrent manager of digital I/Os that is abstracted from platform and hypervisor details to assure reusability, enabling its integration on different system architectures without major changes, simplifying the system design. In addition, it includes a set of measures and diagnostic techniques to assess random and systematic failures. The digital I/O server periodically updates the values of the inputs to refresh the information of the partitions where the inputs are required. Afterwards, the implemented diagnostic techniques are executed, so that, if a failure is detected, the outputs will be refreshed with the safe value instead of with the value provided by the partitions. In the case that different partitions try to update the same output with different values, the partitions will be moved to a safe state and the outputs will be updated with their default value.

The conditions, measures and diagnosis techniques which are implemented by the digital I/O server partition to assess that the digital I/Os controlled by safety-related partitions don't cause a failure that can affect to other partitions are the following:

1. The safety-related outputs have associated inputs with the same or opposite values. The values vary depending on the configuration.
2. The cyclic redundancy codes (CRCs) of the values of the registers associated to the digital inputs and outputs are periodically compared against the values already stored by the digital I/O server. The comparison period is determined by the minimum refreshing period of the digital outputs.
3. Check that the partitions in charge for updating the digital outputs refresh the values of DIOS partition. For that purpose, this solution implements a token that is updated every time that the communication is refreshed, always agreeing with the expected values in DIOS. This solution may also be applied in the remaining partitions, but with the inputs to assure that the communication among partitions and DIOS keeps working.
4. Every time that the values of the digital outputs change, shall be checked that the register values match with the values supported by the DIO Server.
5. Each digital input shall be checked to detect whether their values change. These checks shall be executed under a pre-configured timeout. If the timeout value is not specified, the default value will be used (a month) and the developer will be forced to integrate an output to change the values of the inputs in a controlled non-safety way for testing purposes.

3.2.1.2 COTS device related patterns

Shared memory diagnosis. PAT-SMD-00

The transition from conventional federated architectures to integrated architectures enables the integration of functionalities with different levels of criticality (such as safety, security and real-time) on the same embedded computing platform. This trend is supported by the transition from single-core to multi-core and many-core architectures. Multi-core architectures provide benefits in terms of cost, size, weight reduction as well as improved scalability. However, they imply certification challenges, among others; due on their shared resources (e.g., memory and peripherals) which can lead to interferences in general that can influence the behavior of the safety-related (e.g., in temporal domain).

The sharing of resources is a habitual implementation in today's multi-core mixed devices for improving the performance. These resources can be accessed at the same time from multiple components of the device (e.g., cores and soft-core processors) through regular memory operations and requests. These accesses may cause interferences in general that can imply deviations in the behavior of the system. The IEC 61508 safety standard recommends a set of measures and diagnostic techniques to detect the random failures of variable and invariable memories (see Tables A.5 and A.6 of the IEC 61508-2). However, these measures and diagnostic techniques are focused on single core architectures where, as a general rule, a resource cannot be accessed by more than one component at the same time. Instead, in multi-core architectures, it is common that a resource (e.g., memory or peripheral) can be accessed by two or more components (e.g., two CPUs) at the same time, which may lead to the failure of the system.

This pattern aims to provide a generic diagnostic technique to detect failures in the shared memories of multi-core devices. This solution implements a cyclic redundancy check (CRC) based diagnostic with comparison to detect failures of the shared memory. The application data that is sent through the shared memory is used to calculate a CRC which is stored in memory (e.g., DDR). In addition, a golden CRC of the data that is sent is calculated and stored in the memory (e.g., OCM) by each core. This golden CRC is used to perform the comparison with the CRC value of the data that is sent through the shared memory and determine if the shared memory is source of failures. The calculation of the CRCs can be realized at the beginning or at the end of the execution of the tasks. In the case that the CRCs are calculated at the beginning, a synchronization mechanism may be required to synchronize the calculation and comparison of the CRCs.

Cache Coherency diagnosis. PAT-CMU-00

Cache coherency is the consistency of shared resource data that ends up stored in multiple local caches (e.g., L1 cache and L2 cache). For example, it stores the copies of data saved in several caches. When one copy of data is modified, the other copy shall be changed, otherwise an inconsistency shall arise. Here is where this cross-domain pattern is focused, ensuring that changes of data are propagated through the device and if not, detecting whether a coherency failure occurs. There are three main coherency mechanisms (Directory based, Snooping and Snaffling) which are usually used to provide coherency of memories. In today's mixed-criticality systems based on multi-core devices, the coherency management unit is implemented for managing, among others, the coherency of the processors, the memory and the programmable logic (PL).

This cross-domain pattern aims to provide a generic diagnosis technique that enabled the detection and control of the coherency-related faults proposing the following three possible approaches where the coherency management unit is analyzed from safety perspective.

1. Check the configuration of the coherency management unit
2. Failures caused by unexpected behaviour of coherency management unit (Random faults)
3. Failures caused by external influences (Systematic faults)

3.2.2 VOSYS open-source: Scheduling heuristics and coordination for KVM

In the context of WP2, VOSYS has implemented different scheduling enhancements for Linux-KVM in order to consolidate mixed-criticality systems on top of the KVM hypervisor, while meeting soft real-time constraints of mission-critical tasks. These DREAMS project results have been made available to the community through the DREAMS public repository as follow:

- Paravirt-ops - Coordinated scheduling:** Linux-KVM scheduling has been extended with a coordinated scheduling mechanism based on para-virtualized interface that enables a communication link between the host and guests. Such an implementation allows the host scheduler to be aware of any guest actors in the system. The main goal of the coordinated scheduling is to achieve, among others, low latency and reasonable time to start interactive application and soft real-time tasks since the host scheduler can be aware of guest prioritization. On ARM architecture, the communication between the host and guests can be achieved with the Hypervisor Call (HVC) instruction, which can have arguments to pass different types of information. In this context, each guest can dynamically inform the host when it needs to be prioritized, then the host can adapt the scheduling policy. The corresponding patch series provided by VOSYS have been implemented on the Linux kernel v4.7 available [here](#).

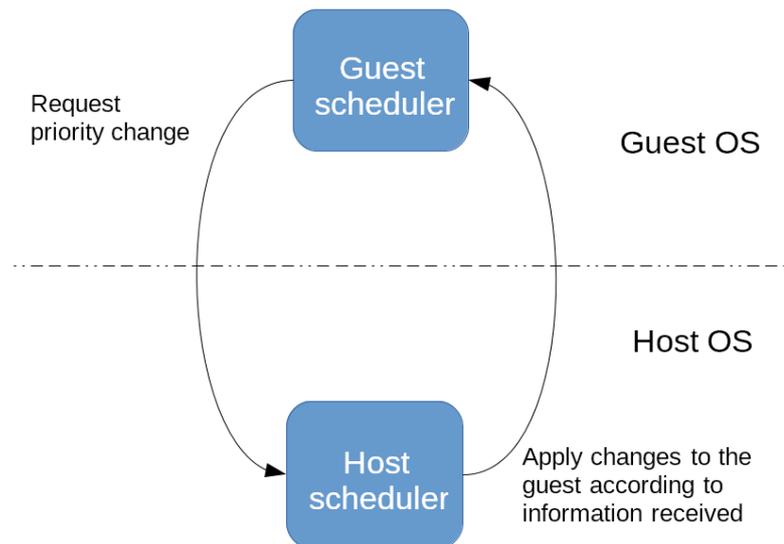


Figure 6: Coordinated scheduling mechanism

- Memguard extensions for KVM guests on ARMv8:** The kernel module Memguard, a memory bandwidth management system developed by Heechul Yun from the University of Illinois has been enhanced to support ARMv8 architecture and to enable memory bandwidth regulation between virtual machines guests, which run on top of Linux-KVM. The main concept is based on the fact that the host scheduler (Linux-KVM) is made aware of the memory bandwidth needed by the guest. Such an implementation has been achieved by using the Paravirt-ops interface, which has been implemented for the coordinated scheduling, to dynamically prioritize the memory bandwidth needs of the guest. Additionally, Memguard has been extended on the ARMv8 architecture to use the Performance Monitor Unit (PMU) in order to access the cache-miss counters, since the memory separation between kernel and user-space is implemented to enable a fine grain control of the memory bandwidth. Finally, an “Idle notifier” chain has been added in order to forward the information about CPU. This [extended ARMv8 Memguard software](#) must be executed in the host with the Linux kernel updated with the paravirt-ops mechanism.

MemGuard

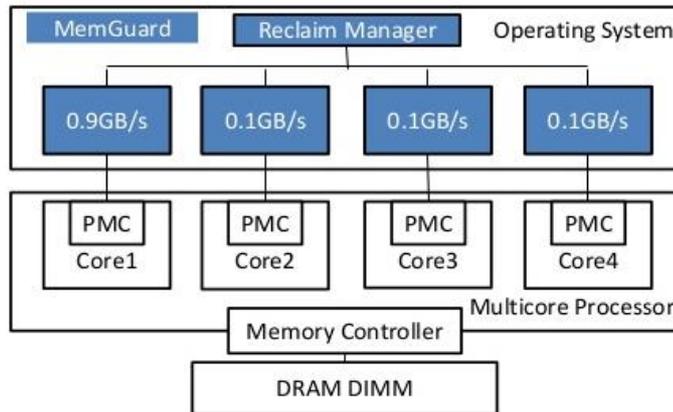


Figure 7: Memguard architecture overview

- Memtalk:** A new mechanism, called Memtalk, has been implemented to enable the communication from guests to the host. Indeed, the guest needs to deliver request messages to the host in order to regulate dynamically the needs of memory bandwidth. The architecture of the solution is split in three main parts: the guest level API to allow user to write/read from a simple file for setting the needed memory-bandwidth value; the hypercall exchange mechanism based on HVC instructions; and parts of Memguard linked to Completely Fair Scheduler (CFS). This kernel module must be executed in the guest in combination with Memguard extended. The software of this Memtalk module for KVM guests is made available at this [link](#).

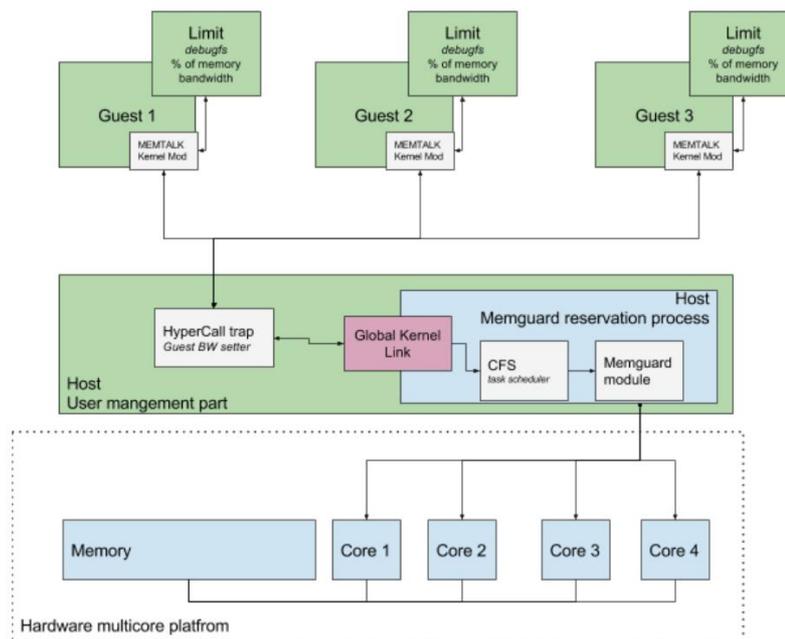


Figure 8: Memguard with virtualization extensions support

3.2.3 TEI open-source: Soft real-time ECG, Mem/NetGuardXt, on-chip security

TEI expects to make contributions to several open source projects. More specifically, four releases to open source community related to TEI work in WP2 are expected in Fall 2017. Refined versions will follow in 2018.

- In relation to distributed embedded soft real-time healthcare solutions, TEI has developed extensions to PhysioNet packages: WaveForm DataBase (WFDB), Open Source ECG Analysis (OSEA) and Waveform Analyzer (WAVE) to perform analysis and visualization on the fly, avoiding excessive re-computation of annotations. This is performed either by adding 3 minutes training to the last signal values, or alternatively, using the last 3 minutes of the pre-trained signal. The contribution will appear on Physionet works, as Soft Real Time ECG Analysis and Visualization. <https://physionet.org/works> This site contains approximately 150 different packages related to ECG processing, but it is one of very few intended also for embedded systems (C+Linux).
- Based on genuine MemGuard technology, TEI has developed two Linux Kernel Modules for memory bandwidth management at CPU-level (by using hardware performance counters to monitor the number of last-level cache misses) and network bandwidth regulation at IP-level (using netfilter hooks to count packet accesses (and throughput) at the incoming or outgoing network interfaces). Both modules (MemGuardXt and NetGuardXt) will be released as open source via sourceforge. Related to NetGuardXt (which can also act as firewall), TEI has released an off-chip network firewall with conflict resolution, see <http://netfirecore.sourceforge.net>
- TEI aims to release an on-chip firewall module as open source hardware. The module has been developed from SystemC models using High Level Synthesis (Vivado tools), while hierarchical linux drivers have been developed to support our healthcare implementation on ARMv7 (Zedboard). The high-level driver currently supports on-chip security, e.g. when a doctor tries to access data from another clinic (or hospital department). More specifically, patient data information is stored initially in BRAM by the administrator, and doctor accesses via hashing to specific BRAMs are valid only from certain network-on-chip input ports mapped to specific Linux groups. Since Linux groups are related to clinics, patient data belonging to a clinic can be accessed only from doctors of the same clinic. In another scenario motivated by automotive technology, we use the NoC firewall to perform multi-compartment protection, i.e. decryption keys for data obtained from an engine control unit (we use ECUSIM2000 simulator) arriving via a CAN bus are stored in BRAM by administrator, whereas hashing is used to dynamically access these keys when CAN bus data must be decoded. Only processes belonging to specific groups can access these keys. In our current prototype on Zedboard, we access CAN bus data via USB (we use the USB2CAN chip); direct access to CAN bus is possible, but requires further hardware support (prototyping in Vivado).

On another axis of research related to WP9 (standardization), TEI has developed and prototyped a new real-time co-simulation methodology for validating correctness and performance characteristics between a hardware IP (DUT) prototyped on an FPGA development board as a full system and its equivalent executable system-level specifications. Inter-module communication between the two domains is supported by an asynchronous channel that bridges SystemC TLM sockets with the real world via either custom, low-cost Linux kernel networking (if system-level simulation runs on a Host PC), or shared memory (if simulation runs on the same development board that contains the DUT). For real-time co-validation, the SystemC clock of the system-level IP is synchronized to the real system clock using a retimefy module. Preliminary evaluation is based on prototyping the NoC Firewall module (DUT) on the FPGA of the ARM v7 Zedboard. The DUT, supported by a hierarchical Linux driver, attempts to thwart malevolent or corrupt on-chip access to BRAM by setting up read/write deny rules that take into account the actual network access path. Preliminary results indicate that co-simulation in ms period range is possible, while it is preferable to place the system-level IP on the board (together with the DUT), rather than on a faster Host PC. The framework and prototype will also be delivered as open source, targeting first a presentation or demo at DATE, FDL or DVCon conference.

Notice that, an interesting cycle-accurate model (gem5 STNoC) developed by TEI in collaboration with ST cannot be released as open source due to restrictions from ST Legal Department. However, the tool has been used within the DREAMS project (USIEGEN, RTAW) for modeling issues related to mixed criticality networks-on-chip. The situation is also similar for the Linux driver of the ST BodyGateway (BGW) device developed by TEI with ST support. Releasing this driver, would be equivalent to revealing details of the BGW hardware architecture.

3.2.4 USIEGEN open-source: DREAMS MCEL

During the lifetime of the DREAMS project, the chair for embedded systems at university of Siegen developed a “Mixed-Criticality Extension Layer (MCEL) for on-chip network interfaces” which supports mixed-criticality by establishing temporal and spatial partitioning in the chip-level communication. At the same time, this work has been published and finalized as a PhD candidate research at this chair.

This building block is attached to the Network Interface (NI) of typical Network-on-Chips (NoCs) and offers time-triggered, rate-constrained and best-effort communication. In addition, the MCEL supports on-the-fly reconfiguration and collects and reports the monitoring data (including statistics and error logs) to the application layer.

This project result has been made available as two open-source projects as follow:

- **MCEL simulation model (MCELSim):** The Gem5¹ simulator provides a flexible and modular simulation system that is capable of evaluating a broad range of systems and is widely available to all researchers. This open-source simulation environment has been chosen as the base-line for the simulation model of the MCEL (known as on-chip LRS in DREAMS and deliverables), as it is worldwide used by the industry as well as the academia. The implemented model of the MCEL uses the NoC of Garnet, which is part of the Ruby network model. This implementation is configurable with a set of CSV files in terms of the topology, communication channels, routes and the communication schedule, when it comes to the time-triggered messages. The textual results of the simulation are highly customizable via the defined *DebugFlags*.
- **MCEL VHDL implementation (MCELite):** The VHDL implementation of the MCEL (known as LRS at on-chip NI in DREAMS and the deliverables) is made available for research purposes. This version has been used in the hardware platform of the DREAMS project (known as DREAMS Harmonized Platform (DHP) in the project documents) and tested with the industrial partners. The VHDL implementation is highly configurable by textual configuration files, whose descriptions are part of the project repository.

¹ <http://gem5.org>

4 Events

Several community events have been organized at different locations, with a strong interaction from the community. The following events were organized in this context:

- Gilching, 13-14 October 2015
- Prague, 18-20 January 2016
- Dresden, 14-18 March 2016
- Grenoble, 20-21 June 2016
- Barcelona, 22 November 2016
- Stockholm, 23-25 January 2017
- Dubrovnik, 27-30 June 2017

These activities focused on the presentation of the project results have been added to the News & Even list of the mixed-criticality forum in order to promote among partners and beyond the DREAMS project (e.g. on the HIPEAC and other project/community websites).

4.1 Gilching, 2015 event

The Embedded Systems Software Engineering Institute (ESSEI)² is a research lab founded in 2012 aiming at improving the development process for complex, reliable and maintainable software intensive avionic systems.

The ESSEI TecDay helped to intensify the exchange between and amongst practitioners and researchers on the latest developments and research in the field of software engineering for avionics. The topics included improvements in the development process, technological issues in hardware and software and the certification process.

The objective of this workshop was to bring together CPU manufacturers, software developers, authorities and researchers from industry and academia to present and discuss recent work in the area of multi-core computing.

The scope of the workshop was both hardware and software aspects of multi-core computing, including system certification aspects, including the following topics

- Multi-core roadmaps of major CPU vendors
- Programming models, frameworks and environments for exploiting parallelism
- Parallel algorithms and applications
- Compiler optimizations and techniques for multi-core systems
- Operating system, middleware, and run-time system support for multi-core
- Correctness and performance analysis of parallel hardware and software
- Tools and methods for development and evaluation of multi-core systems
- Power, energy and performance efficient designs
- Certification aspects of avionic systems based on multi-core

FORTISS contributed the presentation “DREAMS –Distributed Real-Time Architecture for Mixed Criticality Systems” to the workshop. Beyond providing an overview of the project, the DREAMS HW/SW platform and certification strategies, the talk focused on the avionics use case and model-driven engineering methods for mixed-criticality systems.

² <https://www.fortiss.org/en/research/projects/essei/>

4.2 Prague, 2016 event

In conjunction with the 11th HiPEAC conference, which took place in Prague, Czech Republic in January 2016, the 4th MCS workshop was organized. The workshop, which entire title reads “MCS: Integration of mixed-criticality subsystems on multi-core and manycore processors”, was jointly organized by the projects DREAMS, PROXIMA, CONTREX and SAFEPOWER and was focused on solutions for the integration of mixed-criticality subsystems on multi-core processors and virtualization solutions such as hypervisors. Key topics were multicore architectures, software solutions (tools & building blocks) and extra-functional requirements such as energy-power-thermal management, security and safety.

It was scheduled as a 1-day event. After the introduction made by Jon Perez (IKL) that explained the motivation of the research in computer architectures that support mixed-criticality integration and outlined the specific challenges with respect to multi-core processors the major technical results from most representative EC research projects in the field of mixed-criticality were presented within the morning sessions: FP7 DREAMS, CONTREX, PROXIMA, T-CREST and SAFURE and H2020 SAFEPOWER.

During the afternoon two fruitful panel discussions were held. The focus of the first one, moderated by Werner Steinhögl (European Commission), was on platforms, ecosystems and innovation for CPS and started with the vision of the commission on the topic. During the second one, Arjan Geven (TTTech) and Roman Obermaisser (USIEGEN) lead the discussion on Mixed-Criticality Community Building (roadmapping, technical challenges/solutions and exploitation).



Figure 9: Audience of MCS HIPEAC workshop 2016

More than 90 people from industry and academy – representing Europe and Asia – attended the workshop.

The agenda and the workshop material are available at <https://www.hipeac.net/2016/prague/schedule/>

4.3 Dresden, 2016 event

DREAMS participated at the DATE'2016 in Dresden, Germany in March 2016 to present the intermediate integration of the project results. This conference offered an excellent opportunity to meet the experts in the field of mixed-criticality systems.

The DREAMS intermediate integration comprises three dimensions, physical platform (chip/cluster level platform with results of WP2 and WP3), virtual platform (simulation platform with results of WP5) and methodology (meta-models and tools) with a focus on the physical platform.

In addition, the DREAMS Network-on-Chip (a result of WP2) were presented through a talk at IMPAC workshop, whose aim was the latest research results within this spectrum of mixed-critical systems, with emphasis on new on-chip architectures and analysis paradigms to enable fast, yet accurate, and dependable analysis.

4.4 Grenoble, 2016 event

The European Nano electronics Applications, Design & Technology Conference focuses on electronic components, electronic system design, design automation and manufacturing topics related to Micro- and Nano- Electronics. These topics represent very important success factors for European companies. The conference is a showcase of the Industrial state of the art research and development in this area, supported by EUREKA, H2020 and local governments where the latest results and exciting highlights from mainly CATRENE, PENTA, ECSEL/ENIAC, and H2020 projects will form the subjects of this conference

During this event, TEI and ST partners have presented specific results of DREAMS project related to mixed-criticality systems. Indeed, ST has sponsored and contributed to the logistics of the overall event.

4.5 Barcelona, 2016 event

DREAMS with other two projects of the mixed-criticality cluster organized the *Mixed-Criticality Systems Workshop* on November 22th, 2016, which was hosted by Barcelona Supercomputing Center (BSC). This workshop gathered over 70 experts in mixed-criticality systems from industry, academics and the European Commission. Some of the companies and SMEs joining the event include: Airbus, Airbus Defense and Space, Alstom, GMV, Intel, Infineon, Ericsson, Vodafone and ST, among others.

The event was promoted by the European Mixed-Criticality Cluster (MCC) with participation of PROXIMA, DREAMS and CONTREX and EMC2 European projects, whose research is being used in diverse domains such as automotive (cars), avionics (planes), space (satellites) and railway (trains).

In the workshop, each project highlighted the technological developments achieved during the project and also presented some of the selected success stories in terms of technology transfer to market.

4.6 Stockholm, 2017 event

The 5th MCS Workshop took place in Stockholm, Sweden in January 2017 as part of HiPEAC conference. The workshop holds the title “MCS: 5th International workshop on the “Integration of mixed-criticality subsystems on multi-core and manycore processors”. The workshop was jointly organized by various European projects in the field of mixed-criticality i.e. DREAMS, CONTREX, SAFEPOWER, SAFURE, PROXIMA, p-socrates, and Safe4Rail and was focused on solutions for the integration of mixed-criticality subsystems on multi-core processors and virtualization solutions such as hypervisors. Key topics are multicore architectures, software solutions (tools & building blocks) and extra-functional requirements such as energy-power-thermal management, security and safety.

The technical results from most representative EC research projects in the field of mixed-criticality were presented. Moreover, the challenge of the architecture with respect to multi-core processors was discussed in the workshop.

Finally, Arjan Geven (TTTech) and Roman Obermaisser (USIEGEN) lead the discussion on Mixed-Criticality Community Building (roadmapping, technical challenges/solutions and exploitation).

Around 100 people from industry and academy – representing Europe and Asia – attended the workshop.

The agenda and the workshop material can be downloaded from the following link:
<https://www.hipeac.net/events/activities/7443/mcs/#fndtn-program>

4.7 Dubrovnik, 2017 event

The Euromicro Technical Committee on Real-Time Systems (ECRTS)³ promotes state-of-the-art research for applications with temporal constraints, real-time systems. Such computing systems have to provide results which are not only correct, but also delivered in time. Instead of average behavior as for standard computing, real-time systems have to allow for guarantees that the temporal requirements will be met.

The conference addresses research on all aspects of real-time systems:

- Embedded/RT System Design
- Scheduling Design and Analysis
- WCET Analysis
- Power, Energy and/or Thermal Aware RTS
- RT operating Systems and Middleware
- Network/System-on-Chips
- Mixed Criticality Design and Assurance
- Sensor networks
- RT Applications
- Tools and Compilers for embedded systems

In addition, ECRTS is complemented by a series of satellite workshops, which provide lively forums for discussion and interactions on more specific topics:

- Workshop on Operating Systems Platforms for Embedded Real-Time Applications (OSPERT)
- Real-time Network Workshops (RTN)
- Real-Time Scheduling Open Problems Seminar (RTSOPS)
- Workshop on Analysis Tools and Methodologies for Embedded Real-Time Systems (WATERS)
- WorstCase Execution Time Workshops (WCET)

During this event, several partners have presented the DREAMS projects results related to mixed-criticality systems. Indeed, TEI has introduced the full paper “Network and Memory Bandwidth Regulation in a Soft Real-Time Healthcare Application” jointly co-authored with VOSYS and ST in the 13th Workshop on Operating Systems Platforms for Embedded Real-Time Applications (OSPERT 2017). On the other hand, VOSYS has presented the result of the Secure monitor firmware layer, called VOSYSmonitor, through the paper “VOSYSmonitor, a Low Latency Monitor Layer for Mixed-Criticality Systems on ARMv8-A”.

4.8 DREAMS Final Workshop

The DREAMS consortium organized the “Mixed-Criticality Workshop” at the Technical University of Valencia in Spain in the last month of the project life-time to present the latest technological results of the four-year project. The goal of this workshop was to discuss the latest developments in the area of distributed real-time architectures for mixed-critical systems with a group of experts in the domain. This workshop provided the scientists with the latest information on mixed-criticality

³ <http://www.ecrts.org/>

architectures and experience the technological results of the DREAMS project demo sessions, including several live demonstrations of the technologies.

During this workshop the following topics were discussed:

- Presentations and Demonstrations on the Novel DREAMS Concepts:
 - DREAMS Development Methodology: Modelling and Tooling
 - Chip-level, Cluster-level Virtualization for Mixed-Criticality Systems
 - Execution environments for Mixed-Criticality Systems
 - Resource Management in Mixed-Criticality Systems
 - Modular certification and Security in mixed-criticality systems
 - Simulation of hierarchical mixed-criticality systems
- Presentations of the Industrial DREAMS Partners on:
 - Avionics Demonstrator and the Assessment and Demo
 - Windpower Demonstrator and the Assessment and Demo
 - Healthcare Demonstrator and the Assessment and Demo
- Road-mapping and Future Research in Mixed-Criticality Systems

5 Conclusions and Next Steps

This document describes the current activities, updates and the ongoing effort of community building in the project from M24 to M48. As the first milestone of setting up the initial infrastructure was achieved with D9.1.1 and D9.1.2, the community building effort has been focused to provide the right channel in order to share open-source contributions resulted from DREAMS work products. In addition, new projects have been registered to the Mixed-Criticality Forum, thus increasing the impact of the mixed-criticality community by facilitating active exchange of ideas as well as technological building blocks. Furthermore, the MCS community has frequent events that raise awareness of distributed mixed-criticality internationally.

Although the DREAMS project is reaching its end, the community building effort to attract the interest of the industry will continue since many projects registered to the MCF website, which involve DREAMS partners, are still active. Moreover, the DREAMS GitLab infrastructure will be maintained by VOSYS after the project in order to keep available all the open source contributions shared by DREAMS partners.