Distributed Real-time Architecture for Mixed Criticality Systems

Report on intermediate assessment and support for demonstrators
D 1.8.1

Project Acronym	DREAMS	**Grant Agreement Number**		FP7-ICT-2013.3.4-610640	
Document Version	1.0	**Date**	2016-07-01	**Deliverable No.**	D 1.8.1
Contact Person	Donatus Weber	**Organisation**		USIEGEN	
Phone	+49 271 740 3334	**E-Mail**		donatus.weber@uni-siegen.de	

Contributors

Name	Partner
Arjan Geven	TTT
Cristina Zubia	IKL
David Gonzalez	IKL
Donatus Weber	USIEGEN
Hamidreza Ahmadian	USIEGEN
Jörn Migge	RTAW
Roman Obermaisser	USIEGEN
Simon Barner	FORTISS
Tiziana Mastroti	RTAW
Gautam Gala	TUKL
Thomas Koller	USIEGEN

Table of Contents

Executive Summary

This deliverable summarizes the assessment of the intermediate DREAMS architecture in the avionic, wind power and healthcare domains. Based on the assessment, plans of improving the technological building blocks for the final DREAMS architecture are presented.

The technological building blocks with update plans include the DREAMS meta-models, the platform configuration file generators, the on-chip network communication, the composition of schedules, the Safety Communication Layer (SCL), the modular safety cases, the resource management, the virtual platform, the security services and the gateways.

Finally, the demonstrator support activities of the technology work packages are summarized. The goal of the demonstrator support is to promote and support the usage of WP1-5 results by the demonstrators in WP6, WP7 and WP8.

1 Introduction

1.1 Methodology

This deliverable presents D1.8.1 of the DREAMS project. Its main purpose is the consolidation of the assessment from the avionics demonstrators of WP6, the wind power demonstrator of WP7 and the healthcare demonstrator of WP8.

Furthermore a feedback loop is created in order to provide recommendations on how to improve the technological results of the work packages 1-5.

The document consists of the following three major parts:

- Summary of results from the demonstrators (avionics, wind power, healthcare)
- Suggested improvement of technological results based on assessment feedback
- Demonstrator support by WPs 1-5

1.2 Context of the Deliverable

This document is the first output of task T1.8 (Support for demonstrators and consolidation of assessment) in work package WP1. The main goal of this task is to support and promote the usage of the work-package results (e.g., application and platform models) by the demonstrators.

The support includes also help on the identification of opportunities to apply the results. In addition, T1.8 consolidates the assessment from the demonstrators in WP6, WP7 and WP8. The intermediate assessment provides valuable feedback from three application domains and is used to improve the technological building blocks in WP1-WP5.

The final assessment will then quantify significant properties based on the identified metrics (e.g., reliability, timing, fault containment, performance, energy) facilitating the exploitation of the DREAMS architecture.

2 Summary of Results from Assessment

The following subsections provide a short summary of the demonstrator assessment documents D6.3.1 (WP6, avionics), D7.3.1 (WP7, wind power) and D8.3.1 (WP8, health care).

The evaluated results include the DREAMS architectural style and meta-models (WP1), the STNoC (WP2), LRS (WP2), XtratuM (WP2), the secure firmware monitor layer (WP2), resource management with a focus on fault recovery (WP2, WP3), off-chip communication services (WP3), the DREAMS toolchain (WP4), the safety communication layer SCL (WP5), the EtherCAT data logger and fault injector (WP5) and the virtual platform (WP5).

2.1 Avionics Demonstrator (WP6)

The assessment of WP6 is based on the DREAMS harmonized platform (DHP), which has been tailored to fit the needs of the avionics demonstrator use case.

The demonstrator includes three critical and two non-critical applications, while integrating major results from the intermediate DREAMS platform (e.g., Xilinx Zynq, XtratuM, DRAL Hypervisor interface library, DREAMS Local Resource Management runtime library).

The following preliminary assessment targets have been evaluated:

- ***Processor Timing Isolation***: The assessment has investigated the slowdown under stressing benchmarks. In addition, the overhead of the calls from the DLRM library were evaluated.
- ***Multi-core Performance Usage*****:** The multi-core performance was observed in case of parallel applications taking into account the slowdown caused by hypervisor resume and suspend calls.
- ***Multi-core Performance Overhead*****:** The avionic demonstrator has evaluated the intra-partition overhead due to the local resource management runtime library as well as the overhead due to additional partitions for supporting actions of the local resource management.
- ***Fault Management***: The recovery procedure and the ability of recovering from permanent core failures was evaluated.
- ***Applicability***: Recommendations for the tools are provided with a focus on applicability. However, the ability to assess the tools was limited because the interfacing between the different tools was not yet completed.

D6.3.1 [8] provides an analysis of the model-based DREAMS development process, and a preliminary assessment of tools that considered initial versions of tools for configuration, Xoncrete and the model editor AutoFocus3. This assessment complements the initial applicability check of the approach to the avionics use case which has been presented in D1.5.1 [6], Section 4.1.2.1, in the form of preliminary and incomplete models of the demonstrator. In the following, the assessment for the underlying meta-model developed in the frame of WP1 will be summarized.

- As depicted in D6.3.1 [8], Figure 3.13: “Perimeter of DREAMS model”, the DREAMS meta-model is used to describe all relevant aspects of the demonstrator. From the standard PC that serves as an interface for application deployment, experimentation and visualization, only the network interface and the connections to the demonstrator are described.
- The system architect uses the corresponding model editors to define the input models that are subsequently processed by the sub-set of the DREAMS toolchain that is used in the WP6 demonstrator (see D6.3.1 [8], Section 3.5.1).

It is foreseen to apply the tools related to the “Timing Approach” (see D1.3.1 [4]) to the avionics demonstrator, but since most of the scheduling design and verification tools where not sufficiently

ready (M34) at the time of the intermediate assessment, mainly their applicability has been analyzed and reported, see Section 3.5 in deliverable D6.3.1 [8]:

- The *scheduling configuration* tools, especially those able to account for mode changes can be applied, covering partition/task scheduling and on-chip and off-chip communication scheduling.
- The *configuration file generators* for the on-chip and off-chip networks and the multicore scheduling with mode changes are applicable.

In summary, although the tools have not been applied to the avionics demonstrator use case yet, except for modelling, the applicability analysis has led to the identification of gaps that are described in Sections 3.2, 3.4 and 3.5.

More detailed information on the assessment is available in deliverable D6.3.1 [8].

2.2 Wind Power Demonstrator (WP7)

The wind power demonstrator combines safety, real-time and non-real-time functionalities using supervision and control solutions for wind turbines. The wind power demonstrator at this time adopts the DREAMS architectural style and uses the DREAMS core services. The integration of major DREAMS results in the wind power demonstrator is in progress including the harmonized platform, the XtratuM Hypervisor, Windows Embedded CE 6.0, the meta-models and tools, the models (safety, partitions, variability, ...), the modular safety cases, mixed-criticality networks, HW/SW component models, product line validation and certification strategies.

As part of the intermediate assessment, an experimental evaluation of the timing isolation was performed in the scope of the wind power demonstrator. The building blocks Xilinx Zynq 7000, Local Resource Scheduler (LRS), STNoC, PCIe, XtratuM XM-ARM, and DRAL were considered in the evaluation. The observed linear relationship between the WCET overhead and the percentage of overlapping in the applications leads to the recommendation of a limited overlapping between safety-critical and non-safety-critical applications.

Furthermore, a subset of the Key Performance Indicators (KPIs) of the DREAMS project was assessed. The assessment has shown that the timing objectives are met. The project provides most of the building blocks ready-to-be-used as required by the wind power demonstrator. The tools and models provided by the project simplify the development and reduce complexity in many ways, while improving flexibility.

The assessment presented in D7.3.1 does not focus on the applicability of models and tools. However, it states that the application of a sub-set of the tool-set developed in WP1/WP4 (meta-models/offline resource adaptation tools) to the demonstrator is in progress (see [11], Sections 1.4 and 2.4), including:

- XtratuM tool set
- Meta-models (software component model, hardware and system software platform model, safety model, variability model)
- Variability management (BVR), design space exploration (AF3/DSE) and safety-constraint checker

D7.3.1 defines the following Key Performance Indicators (KPIs) related to modelling and tooling (more general categories such as “reduction of development time” will not be listed).

- Percentage of system architecture/design modeled
- Percentage of software application modeled
- Model complexity
- Percentage of development steps covered by tools in demonstrator
- Percentage of development steps potentially covered by tools in wind power

- Percentage of automatically executable transformations
- Effort reduction for addition or modification of features
- Network validation supported by tools
- Percentage of compatible development steps
- Percentage of variability sources successfully handled
- Reduction of variability adaptation time.

As pointed out in [11], Section 3.3, Table 9, almost all of the above KPIs that could be evaluated at the current stage of the project are fulfilled according to plan.

D1.5.1 [6], Section 4.1.2.2 presents early models of the wind power demonstrator use case. These have been refined in D4.3.2 [8], Chapter 4 that also explains in detail the application of the variability management, design space exploration and safety-constraint checker tooling. The refinement of this variability related part of the tool-chain will be followed up in D4.3.3 [9]. For the other parts of the tool-chain that are relevant for the implementation of the physical demonstrator, the gap analysis described in Sections 3.2, 3.4 and 3.5 applies.

More detailed information on the assessment is available in deliverable D7.3.1 [11].

2.3 Healthcare Demonstrator (WP8)

The healthcare demonstrator incorporates the DREAMS Harmonized Platform (DHP) as well as a Juno board using the 64/32-bit ARMv8 architecture. The major technological building blocks that are part of the DHP include (1) the STNoC for chip level communication, (2) XtratuM as the virtualization solution for the ARM A9 dual core and (3) the TTE gateway for the cluster-level communication.

One part of the assessment focuses on the model-based development and tooling including several tools of the DREAMS tool chain (e.g., model editor, design tools, verification tools and configuration tools). It is foreseen to apply the tools related to the “Timing Approach” (see D1.3.1 [4]) to the healthcare demonstrator, but since most of the scheduling design and verification tools where not sufficiently ready (M34) at the time of the intermediate assessment, mainly their applicability has been analyzed and reported, see Section 4.1 in deliverable D8.3.1 [12]:

- The applications that run on client 1 (DHP) and the host can be modelled, as well as the off-chip network that connects them, the on-chip network and the processing tiles in client 1. The Odroid clients, connected via Bluetooth or Ethernet cannot be modelled.
- The *scheduling configuration* tools, can be applied to client 1 (with tiles connected by the on-chip network) and to the off-chip network.
- The *configuration file generators* are applicable for XtratuM, for the on-chip network in client 1 and for the off-chip network changes.

Although the tools have not been applied, the applicability analysis has led to the identification of gaps that are described in Sections 3.2, 3.4 and 3.5.

In the healthcare demonstrator, the virtual platform from T5.2 of WP5 serves for the evaluation of scalability and power/performance tradeoffs of memory interleaving support and evaluating QoS-security tradeoffs when protecting privacy of patient data from malicious processes.

Other assessed technological results are the bandwidth regulation policies at Linux kernel and user-level as well as the scheduling heuristics for KVM. Different scheduling enhancements have been implemented for the healthcare demonstrator. Observed preliminary performance metrics include the interrupt latencies and the startup latencies.

Another evaluation target is the secure monitor firmware layer that is based on the TrustZone security extensions. Its evaluation determines the ability for the concurrent execution of two different operating systems, while ensuring temporal and spatial isolation by means of hardware and software support.

The assessment of the DREAMS off-chip network services is not yet completed, because the interconnection of DHP and Juno using TTEthernet is still ongoing.

More detailed information on the assessment is available in deliverable D8.3.1 [12].

3 Improvement of Technological Results based on Assessment Feedback

3.1 DREAMS Meta-Models

The DREAMS meta-models for platform-specific modelling (PSM; see D1.6.1 [7]) complement the DREAMS meta-models for the application and platform described in D1.4.1 [5] and allow to specify resource utilizations in a tool and device independent format, as well as service configurations of the building blocks of the DREAMS platform. As such, the PSM is the central integration point between

- Different resource adaptation tools developed in WP4 where the resource utilization meta-model serves as an exchange format
- The tool-chain and the hardware/software services of the DREAMS platform: The service configuration model provides a configuration infrastructure that defines the interface to the DREAMS configuration generation framework developed in T4.2, and abstracts the configuration files required for the components of the virtual and physical DREAMS platform.

Because of this integrative role in the DREAMS tool-chain, a number of change requests have been identified to match the resource allocation tools, configuration files and the PSM. They will be explained in the following sub-sections. Due to the mutual dependency between the PSM and the tools and generators, the evaluation of the PSM w.r.t. its applicability to the tools and building blocks of the DREAMS platform has been performed only at the end of the preparation phase of D1.6.1. Hence, it has been decided to shift the delivery of D1.6.1 by one month from M32 to M33, in order to increases the overlap of D1.6.1 with the respective tool deliverables (mainly: D4.1.3 [14], D4.2.2 [15]), and to ensure the coherence of the implementation and the reference documentation in the report (which is essential for the tool integration, see D4.4.1). However, this delay it does not have any impact on the overall time-plan since the latter WP4 deliverables are scheduled for M34.

3.1.1 Schedule Meta-Model Adjustments for On-chip LRS

The following changes are required to make the PSM applicable for the on-chip LRS:

- Mapping the device-independent schedule meta-model (see D1.6.1 [7], Section 3.2) to the configuration model of the on-chip network interface LRS (see also Section 3.4): The schedule meta-model needs to be extended such that the guarding windows for event-triggered traffic as well as the by-pass windows required for TTEthernet and the DDR controllers to access the network-on-chip can be represented. Both windows are determined by the network-on-chip scheduling tool. The required extension involves to define a dedicated *SchedulableEntity* specialization that does not reference a system model element to be scheduled (e.g., component, virtual link), but rather is a placeholder for a guarding or bypass window. That way, an ordinary *ResourceAllocation* can be used to represent the window's phase and duration.
- If a high-priority message arrives while a low-priority message is being processed, the on-chip network LRS needs to apply an integration policy to resolve this media access conflict. In order to handle different possible strategies, the following fields need to be added to the meta-model [13]:
    - `TIMELY_BLOCK`: The switch will not forward any message at times when a time-triggered message is expected.
    - `SHUFFLING`: If a low-priority message is relayed when a high-priority message arrives, the high-priority message is delayed until the processing of the low-priority message has finished (i.e., at most for a maximum-sized low priority message).
    - `PREEMPTION`: If a low-priority message is relayed when a high-priority message arrives, the relay process of the low-priority message is stopped. After the minimum

time of silence on the transmission channel and a delay defined a priori, the high-priority message is relayed.

- In order to match the resolution of the internal representation of times in the physical on-chip network, the representation of times needs to be changed from *double* to *java.math.BigDecimal*.
- In order match the support of the on-chip network's implementation for multiple virtual networks (for each of which a dedicated on-chip network interface is instantiated), a *virtual network ID* needs to be introduced in the meta-model.
- The DREAMS architectural style (see D1.2.1 [3]) defines a dedicated on-chip / off-chip gateway component, which has been reflected accordingly in the DREAMS platform model (see D1.4.1 [5]). However, in the implementation of the DREAMS harmonized platform, the gateway is actually implemented on the ZYNQ SoC's ARM hardcores. In order to the keep the platform model consistent with the architectural style, and still enable the on-chip scheduler to account for this implementation detail, an annotation to the on-chip / off-chip gateway model element needs to be introduced that allows to establish a reference to other platform resources (here: the ARM cores).
- An annotation needs to be introduced to identify the *Node* in the platform model that represents the DREAMS harmonized platform (and for which an on-chip network schedule is required).

3.1.2 Reconfiguration Meta-Model Adjustments for MCOSF tool

The following changes are required to align the reconfiguration meta-model (see D1.6.1 [7], Section 3.3) with the MCOSF tool (see D4.1.3 [14]):

- Ability to specify if a *LocalConfiguration* represents a continuous or a transition mode:
  - A continuous mode is one which is executed cyclically until there is a switching event (e.g., core failure).
  - A transition mode is one which is executed at most once and switches to a continuous mode when there is a safe point to switch. Safe points are where the black-out parameter of a time slot is false (black-out property is defined by the scheduler).
- Ability to specify if a *ResourceAllocation* represents a black-out slot (black out slots are used to define safe points in time to switch modes / configurations).
- Introduce the possibility to model the configuration / plan switch delay of the hypervisor.
- Change semantics of configuration switch *Transition*: Transition is performed if **all** of the referenced resources fail.
- In order to be able to represent the software tasks used to implement the resource management framework, an annotation for logical components needs to be introduced that enables to distinguish the following component types:
  - Regular application.
  - Monitoring component.
  - Local Resource Manager (LRM) component.
  - Global Resource Manager (GRM) component.

3.1.3 Application and Platform Meta-Model Adjustments for Timing Analysis

The following changes need to be applied to the DREAMS application and platform model in order to integrate the timing analysis (see also: Section 3.5):

- In order to enable the timing analysis to know which schedules run on synchronized clocks, the DREAMS platform model (see D1.4.1 [5]) needs to be extended with the following annotation to express clock domains (clocks in the same domain are assumed to be

synchronized): `eu.dreamsproject.platform.model.annotation.ClockDomain` (registered for `eu.dreamsproject.platform.model.tile.Clock`)

- In order to increase the usability of the model, it should be possible to directly annotate messages sizes. Currently, the message size is derived from the logical port's type. Hence, currently, for larger message sizes, it would have been necessary to declare a dedicated data type corresponding to an array of bits of the respective size.

3.1.4 Virtual Platform Configuration Model Adjustments

For both the configuration model of the virtual on-chip and off-chip platform, additional parameters have been identified:

- Buffer size and queue length for ports of the virtual and physical on-chip network interface.
- Queue length for all per-virtual link configuration items of the virtual off-chip simulation components (nodes without a NoC, switches, gateways).

3.2 Platform Configuration File Generators

One of the goals of the DREAMS project is the model-driven generation of configuration files, i.e., an automated generation of configurations files for platform building blocks, out of a verified system configuration, so that no errors may be introduced through "manual" copy & paste. This approach is foreseen for the on-chip and off-chip communication, as well as for the partition and task scheduling.

3.2.1 Configuration Generation Framework

Task T4.2 contributes a model-to-text generation framework based on Acceleo that will be used to implement the configurations file generators for the DREAMS virtual platform, as well as a generator that produces text files that will be translated into the binary configuration required for the physical on-chip LRS.

During the implementation of the configuration generators, a shortcoming of the Acceleo compiler has been detected that leads to erroneous compilation results in case the input meta-model has nested sub-packages (as does the DREAMS meta-model). The problem is addressed by a modification to the Acceleo compiler that will be ready in M34. Furthermore, it will be checked if these modifications to the compiler and further minor fixes to the Acceleo development environment can be contributed back to the Acceleo open source community.

3.2.2 Resource Management Configuration

The partition and the task scheduling are realized by different platform building blocks. In the case of the avionics demonstrator for example, these are respectively XtratuM and the DLRM (DREAMS Local Resource Management). These configuration files need to be consistent with each other. In order to manage the consistency during the initial phase where the model-driven configuration for XtratuM and the DLRM were not available, an intermediate format has been created: PCF (Platform Configuration File). It covers both the partition and the task scheduling. A dedicated tool generates, out of a PCF file, a consistent set of configuration files for XtratuM and the DLRM. For the latter, it includes some C header files for LRS, LRM and GRM.

In order to make model-driven configuration-file generation a reality also for resource management, it has been decided to add plugins to AutoFOCUS3, which generate the PCF files out of a system configuration described according to the DREAMS meta-model. It will be implemented by M34.

3.2.3 Xtratum Configuration

An initial version of the configuration generator for the XtratuM hypervisor has been presented in D4.2.1. It is capable to generate the configuration file for a hypervisor instance that is running on a standalone DREAMS tile. In the course of D4.2.2, the generator will be extended to configure the routing of partition ports via the STNoC-based network-on-chip, by translating a resource-utilization model that contains the schedules and virtual links of the system to the network-interface configuration of the updated version of the XtratuM configuration schema (see D1.6.1). Further, the XtratuM configuration generator will be extended to configure the routing of partition ports via the TTEthernet network (similar to the on-chip network).

3.3 Chip-Level Software (XtratuM)

The XtratuM hypervisor has experienced multiples evolutions throughout the project. These evolutions are the result of the collection of requirements from end users. These end users include the application developers (demonstrators), software providers (resource management, guest O.S. or runtime) and designers of configuration tools (meta-models and configuration file generators).

The hypervisor development uses the assessment results of the demonstrators as a guide to evaluate the improvements already introduced in earlier stages of development and for the identification of new services required for the final integration of the application.

XtratuM is being used on all demonstrators in the DREAMS project through the DHP based on the multicore processor ARM Cortex A9. This platform allows both on-chip and off-chip communication. The former is provided by means of the STNoC using a Network Interface (NI) extension called LRS. In addition, the DHP provides an interface via the NoC towards a TTEthernet controller. In order to support these devices, software drivers were incorporated into the XtratuM kernel. Further, these new communication components led to the extension of the scheme of communication in XtratuM. The scheme is based on communication channels according to ARINC 653 and was extended to allow binding virtual communication ports with physical communication ports. These modifications had impacts on the hypervisor including the internal communication design, the ARM Board-Support Package (BSP) and the specification of the XM configuration file.

In addition to the DHP, the XtratuM hypervisor has also evolved to other architectures such as Intel X86 and PowerPC. In the case of Intel, the XM hypervisor adds support for hardware virtualization extensions available in the Galileo platform of the wind-power demonstrator. This support allows improving the performance and the compatibility of the hypervisor with guest operating systems, such as WinCE 6.0. In the case of the PPC, new XtratuM BSPs have been included to support the T4240 processor. This porting takes advantage of the hardware virtualization extensions to provide full virtualization in the avionics application. The XtratuM version for the PPC must also incorporate a TTEthernet driver and perform bindings between the virtual and physical communication ports. This version includes the improvements of the XtratuM version for the DHP.

The intermediate integration phase for the hypervisor focused mainly on the adaptations of XtratuM to support the hardware required in each industrial demonstrator, on the analysis from a functional and performance point of view, and on the evaluation of temporal interference due to multicore platforms. In the final integration phase, the hypervisor development will be focused on performance improvements for the DHP and Galileo platforms. Also, the complete integration between TTEthernet and XtratuM in the DHP platform is focused on. This integration requires the availability of the final version of the TTEthernet controller and the complete integration of the hypervisor with the T4240QDS platform.

These activities will be performed until end of M34 and the integration of TTEthernet and XtratuM could proceed until M35 depending on the availability of the TTEthernet controller in the DHP.

3.4 On-chip Network Communication

In the course of the project it has been decided that the on-chip/off-chip gateway of the DHP is implemented in the ARM tile, whereas the off-chip network is connected to another tile. This induces traffic over the on-chip network, between the gateway and the off-chip network, which is not modelled as virtual links. In order to integrate this traffic and avoid interference with the safety critical VLs, so-called *bypass windows* have been introduced, where the network is reserved for the communication between the on-chip/off-chip gateway and the off-chip network. These bypass windows need to be taken into account

- by the platform specific model (see Section 3.1.1),
- by the scheduling configuration tool (i.e., RTaW-Timing and On-chip TT Scheduling),
- by the timing analysis tool (i.e., RTaW-Timing and Evaluation),
- by the configuration file generator for the on-chip network communication.

These extensions will be implemented until end of M34.

3.5 Composition of Schedules

In order to tackle the overall scheduling problem, it has been decided in WP4 to decompose the initial scheduling problem into three sub-problems that cover each a different scheduling domain:

- Partition/tasks scheduling.
- On-chip network communication scheduling.
- Off-chip network communication scheduling.

Furthermore, time-triggered scheduling schemes are used for safety-critical tasks and their communication. Together with synchronized clocks (services foreseen by the DREAMS architectural style) it has become possible to align the time-triggered schedules of all three scheduling domains, thereby reducing the inter-domain delays and therefore also the end-to-end delays.

However, the coordination of time-triggered schedules require two properties:

1. The clocks that drive the different time-triggered schedules must be synchronized
2. The periods of the different time-triggered schedules must be harmonic

In the course of the project it appeared that the first assumption is not necessarily true for all parts of the demonstrators, e.g., in the Galileo node of the wind power demonstrator. Furthermore, the implementation of the on-chip network interface uses a time resolution that is a particular fraction of 1 second, namely a negative power of 2. This implies that it is not possible to configure a communication period of exactly 50 ms, as used for tasks in the demonstrators. The solution is to choose a smaller period among those that are possible such as 31.25 ms (=32Hz). With these periods, task and on-chip packet schedules drift against each other at run-time, i.e. no alignment of the on-chip network and the tasks schedule is possible. It implies a “transition” delay equal to one period in the second schedule.

The above observation means that the granularity of clocks and the fact that not all clocks might be synchronized need to be taking into account:

- in the DREAMS meta-model for the clock domains,
- by the timing decomposition tool (i.e., RTaW-Timing / Decomposition),
- by the timing analysis tool (i.e., RTaW-Timing / Evaluation).

These extensions will be implemented until end of M34.

3.6 Safety Communication Layer (SCL)

The Safety Communication Layer (SCL), which implements the safety measures (techniques) defined by IEC 61784-3-3 and developed in T3.3, will only be integrated in the wind power demonstrator. It

has not been done yet. It will be used to transport safety-related input/output data between EtherCAT slaves and the safety protection system deployed in the harmonized platform. An evaluation of the safety features of the developed communication layer will be performed and and later be stressed in the evaluation plan by means of the real-time fault injection framework developed in T5.2. A data-acquisition system (the Data logger developed in T3.4) will be used in order to analyse the safety frames interchanged among the EtherCAT slaves and the master. The whole picture is shown in Figure 1 below.

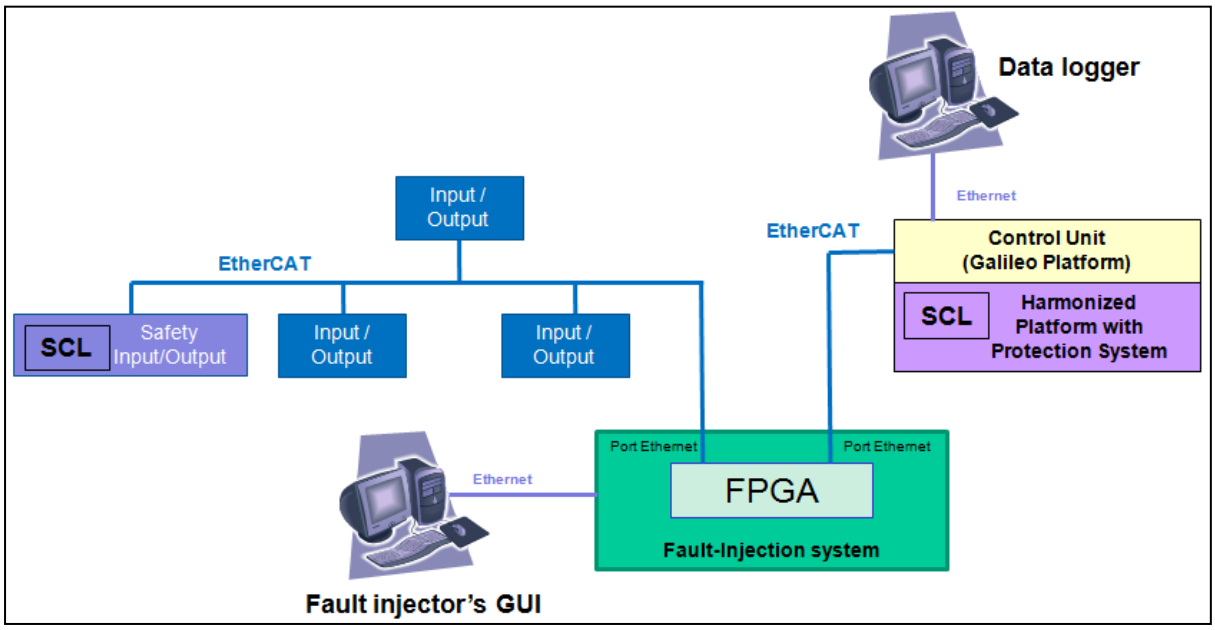

Figure 1 SCL assessment in the wind power demonstrator

The level of integration of the different technologies that take part in the SCL assessment within the wind power demonstrator is as follows:

Technology	Expected at	Status	Comments
Safety Communication Layer	M26	Not started	SCL integration not started either in EtherCAT slave node nor in the harmonized platform. The technology is almost ready but needs some adaptations for the specific needs of the wind power demonstrator.
EtherCAT Datalogger	M33	Not started	Datalogger software is ready but the integration in the demonstrator has not started yet, since the EtherCAT communication is still not running.
EtherCAT fault injector	M36	Not started	Fault injector development is finished. However, integration with the demonstrator has not started yet, since EtherCAT communication is still not running.

Table 1 Integration level in the wind power demonstrator. (Source: D7.3.1 [11])

The assessment of this communication block within the wind power use case is being tackled in T7.3. No feedback has been received yet as the integration of the technology, as said before and shown in Table 1, has not been started yet. But as the SCL has already been evaluated in experimental setup using a RS-232 communication layer, no big issues are expected related to technology improvements.

The assessment result will be used to improve the fault injector and the data logger systems as well.

3.7 Modular Safety Cases

A 'safety case' "*represents an argument supporting the claim that the system is safe for a given application in a given environment*" [14]. It provides I) arguments to demonstrate that safety properties are satisfied and risk has been mitigated, II) a notation mechanism that is often required as a piece of the certification process and III) interoperability among different standards and domains (e.g., avionic, automotive, railway).

A well partitioned safety case limits the impact of changes to a reduced area of the safety case and enables the reusability of these parts. Partitioning is a complexity management technique [15] that subdivides the system into smaller parts (modules) that are independently generated and used to compose the system. On this basis, the implementation of modular safety cases potentially enables the reusability of predefined modules, reducing the overall complexity (simplification strategy) and supporting the limitation of change impacts to specific modules.

Modular safety cases are defined from the perspective of the 'system architect', in order to ease the specification, development and certification of mixed-criticality product families composed of a large number of 'building blocks'. FP7 DREAMS (WP5) contributes with the definition of modular safety cases for selected 'building-blocks' of mixed-criticality systems: safety hypervisor and partition(s) (D5.1.1), multicore COTS processor(s) (D5.1.2) and mixed-criticality network(s) (D5.1.3).

IEC-61508 is selected as the safety reference standard because results could potentially be extended in the future to different domain specific standards that take IEC-61508 as a reference standard (e.g., railway, automotive, vertical transportation, machinery…). Within the DREAMS project this standard only applies to the wind power demonstrator.

The different modular safety cases will be integrated within the wind power demonstrator by means of the toolchain developed in WP4, where recommendations on how to proceed with the integration and the needed documentation to face a certification process will be suggested. Figure 1 illustrates the modular safety-cases process. :

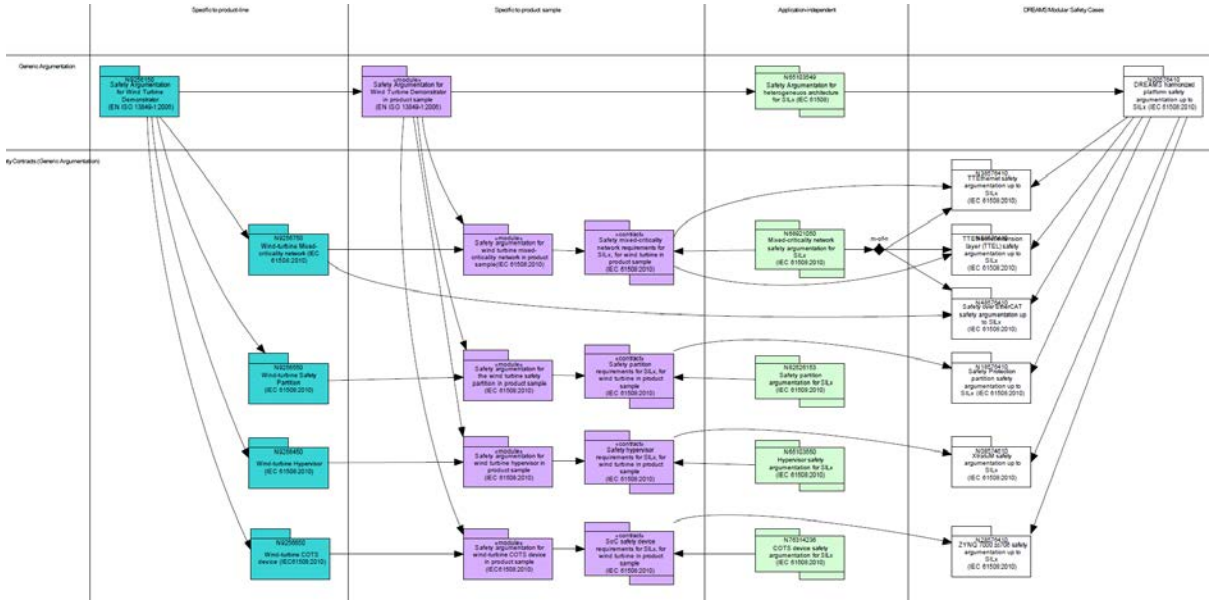

Figure 2 Example on how to precede with the modular safety cases

The level of integration of the modular safety cases in the wind power demonstrator is as follows:

Technology	Expected at	Status	Comments
Safety Communication Layer	M26	Not started	SCL integration not started either in EtherCAT slave node or in the harmonized platform. The technology is almost ready but needs some adaptations for the specific needs of the wind power demonstrator.
Modular safety cases for hypervisor	M22	In progress	Almost completed, waiting for WP4 tools.
Mixed-criticality network	M26	In progress	On-chip network is already implemented (though not tested). Off-chip network is not implemented yet.

Table 2 Integration level in the wind power demonstrator. (Source: D7.3.1)

The assessment of the modular safety cases has been done in two different phases. The first assessment by the certification authority TÜV Rheinland was done for each modular safety case, where the wording was updated based on the recommendations. In this second phase the assessment will be done within the wind power use case. There is no direct feedback from the demonstrator development but the modular safety cases have been integrated as support documentation to the WP4 toolchain without problems.

A more detailed feedback is expected when the development of the use case is finished. Changes in the wording of the modular safety cases are not expected but the integration within the toolchain will be refined.

3.8 Resource Management

The Resource Management services (RMS), defined in D3.2.1, and developed in T3.2, T2.2 and T2.3, will only be integrated in the avionics demonstrator. The integration is currently in progress. The RMS

in the DREAMS platform are realized by a Global Resource Manager (GRM) in combination with local building blocks for resource management: Local resource manager (LRM), Monitor (MON) and Local resource scheduler (LRS). All the applications of the Avionics demonstrator are under control of the LRM, which schedules the execution of the applications tasks via the LRS, and monitors their performance and the total slot execution time (i.e., including the time performing LRS actions) using the MON capabilities. Additional, MON partitions are also deployed for core failure detection. During each Major Cycle (MaC), the LRM collects the monitored statistics, and applies a local reconfiguration, if necessary. If a problem cannot be overcome by the LRM locally, then it sends a reconfiguration request to the GRM. The GRM then tries to resolve the issue by performing a global reconfiguration.

The Global and Local RMS are implemented as part of the DLRM library. Functionality for secure communication between LRM and GRM are also incorporated in this library. The integration of the DLRM library is a complex task. The resource management services have been integrated in the avionics demonstrator to support two types of failure: 1) permanent core failures leading to local and global reconfiguration; 2) temporal overload situations when best-effort applications perform excessive accesses to shared resources, thereby leading to reconfiguration. RM services are integrated in the avionics demonstrator as follows:

- The GRM runs as a critical application in the harmonized platform. In the context of the DREAMS project, it is assumed that the GRM runs on a fail-safe core.
- The LRM runs as a critical application and executes at end of every MaC. A LRM is present on each core. All LRM are synchronized, and one amongst them is a master.
- The LRS is dispatched at the start of each user-application slot to schedule the tasks.
- A MON partition must be present on each core executing in each MaC to monitor the core's health status.
- An update or reconfiguration message must be sent by the LRM to the GRM every MaC. These messages also act as membership messages. If the GRM fails to receive a message every cycle, then it considers the node corresponding to the LRM to be dead. Additional membership channels will not be used.
- The MON gathers the information at runtime from the system and the application may be integrated within the application partitions.

The status of the RM services is shown in Table 3.

Technology	Expected at	Status	Dependencies	Comments
GRM	M36	In progress	DRAL, T4.1 (global reconfiguration graph), harmonized platform	GRM is being developed as part of the DLRM library. The technology is almost ready. It needs some adaptations for integration in avionics demonstrator. The GRM will run only on the harmonized platform.
LRM, MON, LRS	Depends on availability of DRAL for T4240 and TTE gateways.	Not started yet	DRAL, TTE gateways, T4.1 (Local reconfiguration graphs), harmonized platform	Local RMS are being developed as part of the DLRM library. The LRM implementation for the harmonized platform is ready and integrated in the Avionics demonstrator via the DLRM library. The core failure

				capability has been integrated into the DLRM and successfully tested on the ARM platform. The deadline overrun management and QoS capabilities are currently under active development DRAL is not yet available for T4240. Hence, LRM, MON and LRS have not yet been developed for this board.
Secure communication between GRM and LRM	Available	completed	DRAL, TTE gateways, security services, harmonized platform	The technology is ready and implemented in the avionics demonstrator as part of DLRM library. In depth evaluation has not yet been performed as LRM is not yet available for T4240.

Table 3 Integration of RMS in avionics demonstrator

All the partners involved in the development are working closely with the developers of the avionics demonstrator (TRT) for integrating the DLRM library with the applications in the demonstrator. Bi-weekly telephone conferences with all involved partners are organized to update the technological results based on the assessment in D6.3.1, and to ease the integration efforts. In the upcoming months, we are further enhancing and debugging the DLRM library for better integration with the demonstrator.

3.9 Security Services

The security services for cluster level communication are separated into three parts: the security services for off-chip communication, the secure time synchronization and the security services for application level communication.

The off-chip communication security services are implemented in the TTEthernet gateways and in the TTEthernet switches (“off-chip routers”) as shown in Figure 3. These services provide a secure data transmission, i.e., confidentiality and integrity for each link.

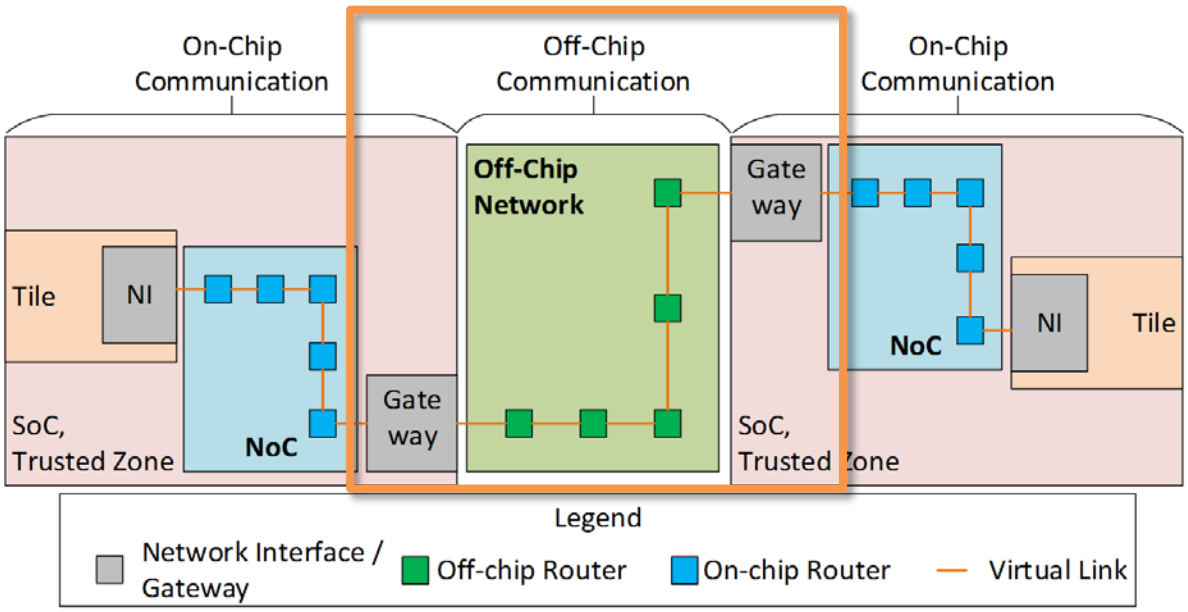

Figure 3: Secure off-chip communication

A MACsec implementation for TTEthernet is used to provide these services. The assessment feedback from the demonstrators showed that the computation time for MACsec has to be reduced compared to the first implementation. This will be solved in the next version. In addition, there are improvements for the secure time synchronization.

Security services for application-level communication were implemented to secure the communication amongst resource managers. These services are implemented in a security library. The security library provides the generic security services confidentiality, integrity, authenticity and access control on application level. The security library is implemented logically as a security layer between a resource management component and the XtratuM hypervisor as shown in Figure 4.

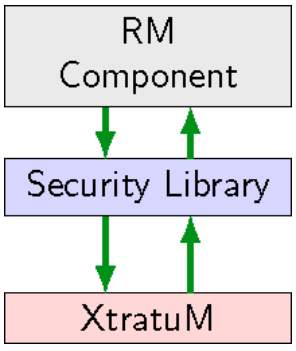

Figure 4: Security library

During the development of the security library there was already helpful feedback from the avionics demonstrator. Based on this feedback, the security library is implemented in such a manner that it also supports a secure communication between applications. This option will be used in the avionics demonstrator for different applications in the system.

3.10 Gateways

3.10.1 Gateways in the Avionics demonstrator

Deliverable D6.3.1 [10] indicates the intended use of the gateway services and their abstractions through the DRAL in order to distribute the applications between several nodes. In the final evaluation, "the integration of the GRM and LRM communication on the off-chip network will be started and the evaluation of the distributed system assessment aspects of the DREAMS framework will take place" [10]. In the first phase of the evaluation, the main focus of the avionics demonstrator has been to look into the overall DREAMS concepts applied on the local level and has not focused on the, more complex, distributed nature that is made possible through the DREAMS gateways, also because the software integration in the hypervisor for the DHP was not available at the point where the intermediate evaluation took place.

In the next phase of the project, the integration through the network will be evaluated, as displayed in the following figure:

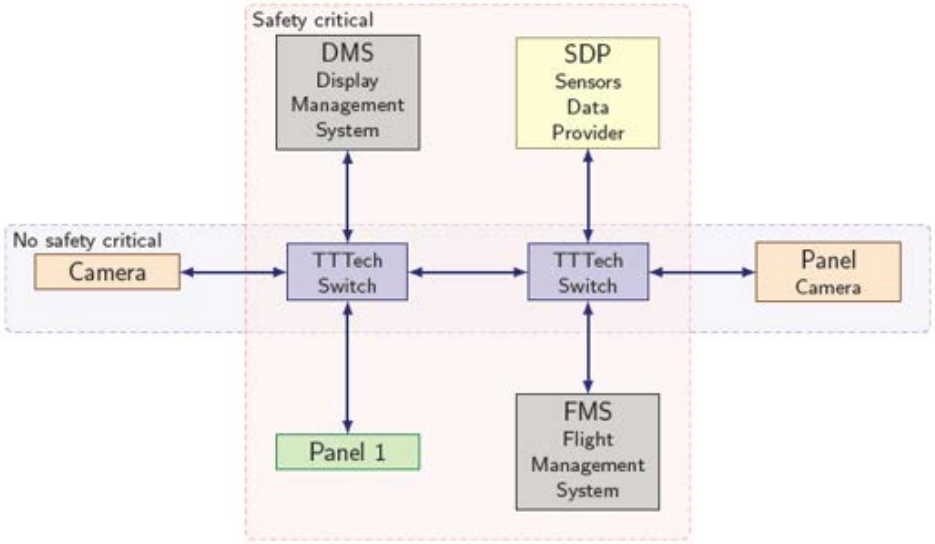

Figure 5: Networked mixed-criticality avionics demonstrator with TTEthernet network.

The gateways and off-chip network will be used to securely connect the different components together using the DREAMS abstraction layer and respective communication services.

3.10.2 Gateways in the Wind power demonstrator

The wind power demonstrator utilized the gateways from WP3 in order to connect between the Galileo PC and the DHP, in order to exchange data between the EtherCAT ring and the harmonized platform, thus allowing safety-relevant values to reach their destination. The Control partition on the Galileo owns the EtherCAT interfaces and is responsible to forward relevant data through the PCIe.

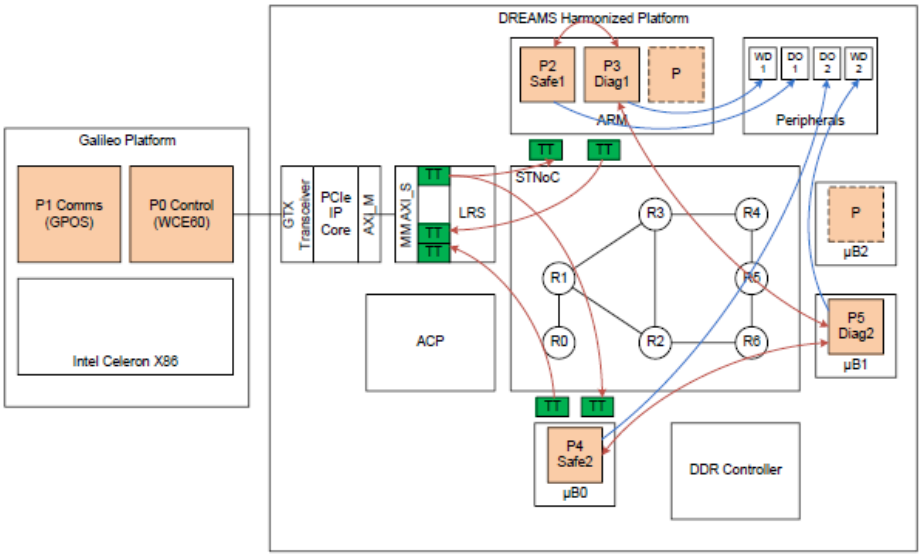

Figure 6: Mixed-criticality wind power demonstrator with EtherCAT network

The architecture and detailed design for the wind power demonstrator off-chip communication have been completed, and the hardware implementation is close to finalization. However, the use of the gateway software and specifically the Safety Communication Layer has not been evaluated yet as the SCL integration was not started in the EtherCAT slave node and harmonized platform at the moment of the intermediate evaluation. For this reason, the evaluation for the gateways could not yet be completed. The technology is almost ready but needs some adaptations for the specific needs of the wind power demonstrator and will be evaluated in the final phase.

3.10.3 Gateways in the Healthcare demonstrator

The healthcare demonstrator uses the gateway to facilitate access through TTEthernet, combining time-triggered, rate-constrained and best-effort data. The healthcare demonstrator utilizes the mixed-criticality network in order to communicate between the DHP and the Juno board (both critical and non-critical traffic) and the ODROID device (only non-critical traffic).

In order to support both DHP and Juno platforms, integration with the respective hypervisors is a necessity. On the DHP, the used hypervisor is XtratuM. On the Juno board, the hypervisor is KVM. The Odroid device does not require DREAMS-specific services.

3.10.4 Gateways Next steps

In order to accommodate for optimal results in the final demonstrator evaluations, the following activities are performed:

- Final release of **gateway firmware** and **switch firmware** to provide support for the off-chip communication with:
  - Complete transparency of communication security services through TTEthernet for the healthcare and avionics use cases
  - Full support for reconfiguration scenarios for the avionics use case
  - Full support of the services that are defined in the gateway requirements and design documents.
- Final release of **gateway embedded software** implementing the off-chip gateway services support in both XtratuM and KVM in order to support the avionics, wind power and healthcare use cases.
- Final release of the **Configuration tools** for gateway and network
- Final release of the **Safety Communication Layer** software
- Final release of the **Security Services library**

4 Demonstrator Support

This section summarizes the demonstrator support activities in WP1-5.

4.1 WP1

The support provided by WP1 is mainly focused on the DREAMS meta-models (elaborated in Section 3.1) and the DREAMS Harmonized Platform (DHP) which is identified together with all DREAMS partners for bundling integration efforts and effectively supporting the strong role of technology partners (cf. D1.5.1 [6], Section 2.1).

The DHP is realized using a Xilinx Zynq-7000 AP SoC ZC706 evaluation kit, which is composed of a Processing System (PS) and a Programmable Logic (PL). The PS contains the application processor unit, input output peripherals, interconnect to the PL and memory interfaces. The PL provides a rich architecture of user-configurable capabilities, such as, configurable logic blocks, block RAMs, configurable I/Os and integrated interface blocks for PCI Express designs.

When it comes to the memory, the ZC706 evaluation kit offers 1 GB DDR3 component memory at the PS side. In addition to the memory offered at the PS side, the board offers 1 GB DDR3 SODIMM memory on the PL side which can be used by developed hardware components of the FPGA.

After the intermediate integration, based on the assessment report provided by the DREAMS demonstrator partners, improvements in the DHP were carried out which are described in this section.

4.1.1 Local DDR Access at the ARM Processor

As described in D1.5.1, STNoC connects the ARM processor and the MicroBlazes to the DDR controller and the TTE gateway controller. Initially only the DDR controller at the PS side was planned to be used by the network-on-chip. This means that the application running on the ARM processor would be stored on the DDR and each read and write would have to go through the STNoC, which implies a significant performance drawback.

However, based on the requests from the demonstrator partners, one fourth of the PS memory which is accessed by the cores through the STNoC was replaced by the DDR controller at the PL side, in order to release the replaced memory to be accessed locally by the core. More precisely, initially direct accesses to the local memory at the PS side was blocked, to restrict the access to the memory only through the STNoC; but thereafter, one fourth of the memory at the PS side was dedicated to the application running on the hypervisor to be accessed locally. In order to still offer 1 GB of memory by the interconnect, the equivalent amount of memory was used from the DDR at the PL side (see Figure 7).

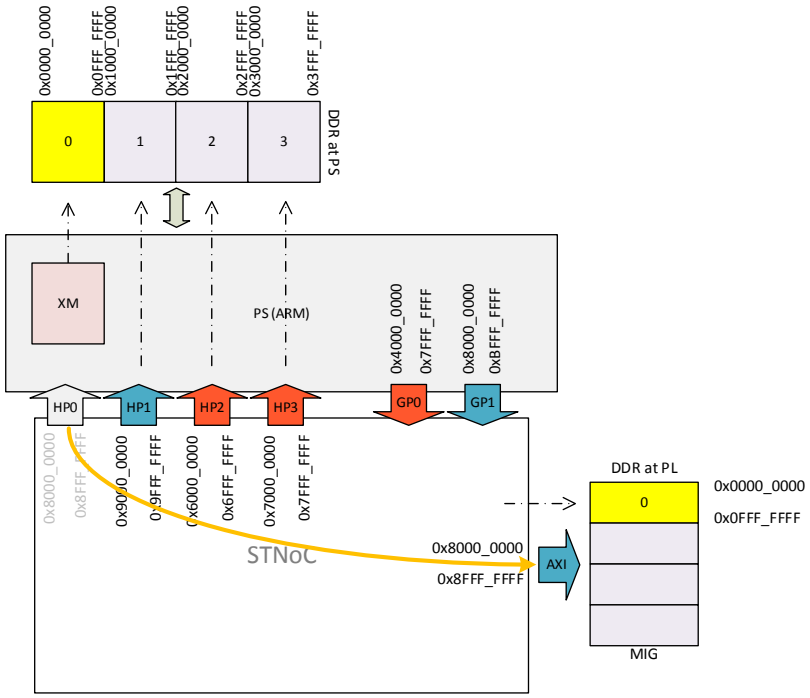

Figure 7 Use of the memory at the PL and releasing one fourth of the memory at the PS to be accessed locally

4.1.2 PCIe interface for the Wind Power Demonstrator

As described in D7.1.1, the platform of the wind power demonstrator contains the Galileo platform and the DHP. In this set-up, the Galileo platform is on the one hand, connected to the DHP via a PCIe connection, and on the other hand, connected to the input/output units and the protection systems via the EtherCAT (cf. Figure 8). Hence, there was no need for the TTE gateway in the DHP for the wind power platform.

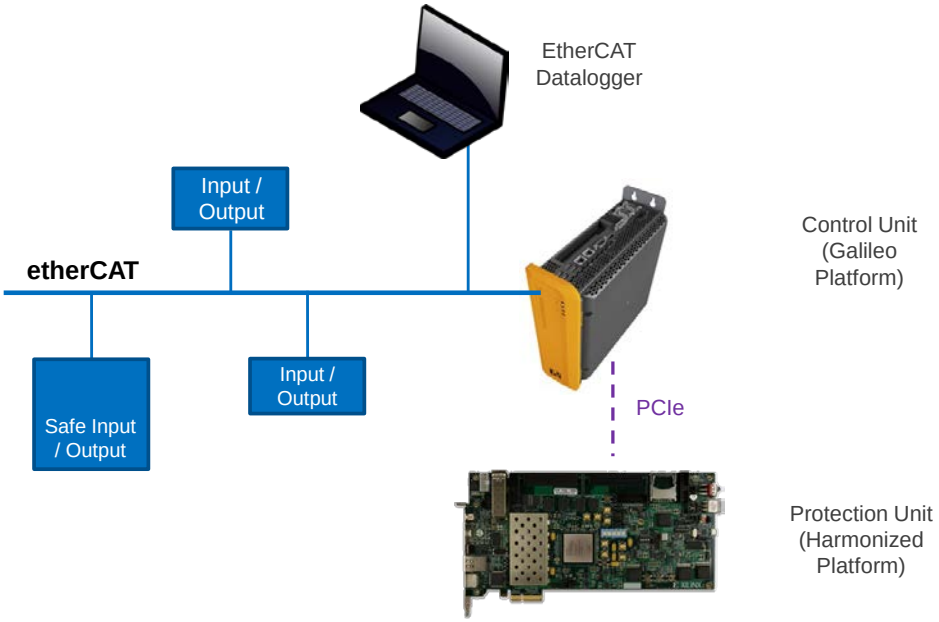

Figure 8 Wind Power Use Cases

Based on the request from the wind power demonstrator, the TTE controller was removed from the DHP and an AXI PCIe controller was added instead. In order to achieve temporal and spatial

segregation, the PCIe controller could be connected to the STNoC, but through an on-chip LRS (cf. Figure 9). In this case, the PCIe controller acts as an AXI master and initiates read and write transactions targeting the LRS, for writing the messages into or reading them from the LRS ports.

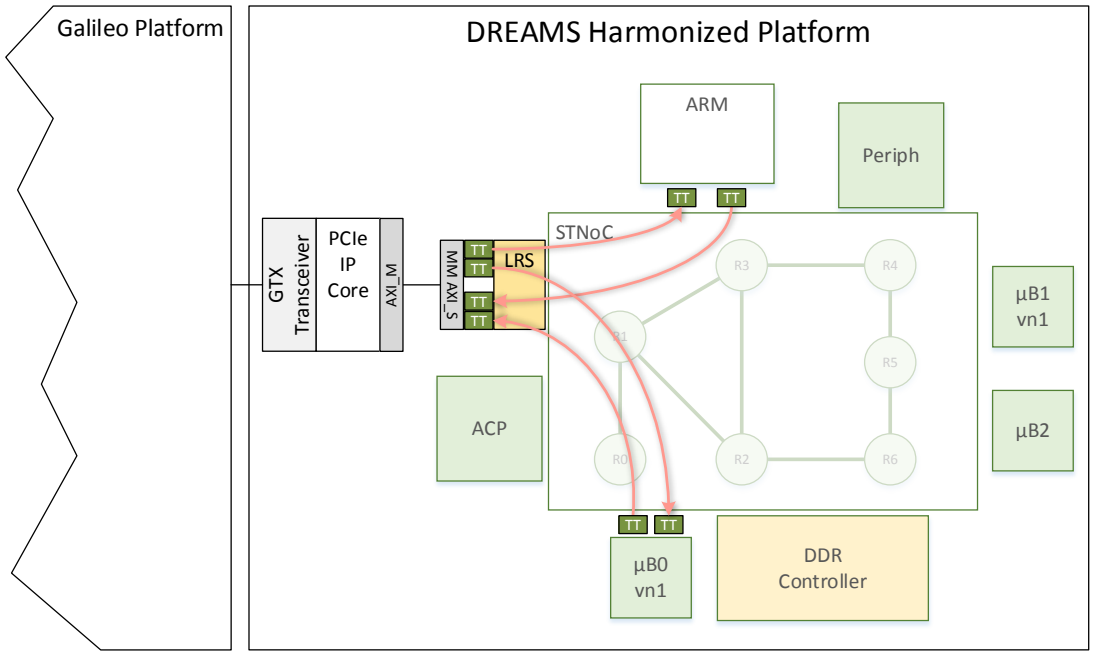

Figure 9 On-chip LRS between the STNoC and the PCIe IP core

4.1.3 Incoming Traffic to the ARM Tile

STNoC serves as the heart of the network-on-chip and offers the processors a memory-mapped access to the peripherals, the TTE controller and the DDR controller. In the initial scenario, there would be no transactions which target the ARM processor and hence the Network Interface (NI) at the ARM processor was not capable of receiving transactions targeting the ARM. In contrast, as there is no transactions initiating from the peripherals, the NIs dedicated to the peripherals were not capable of initiating transactions (left figure in Figure 10).

After addition of the LRS to the NIs of the STNoC and adding the message-based communication between the cores, bidirectional communication at the ARM processor was required. Taken into account that the NIs used for the peripherals of the board were not used anyway, those NIs were connected to the LRSs at the ARM processor to provide bidirectional communication for those LRSs (right figure in Figure 10).

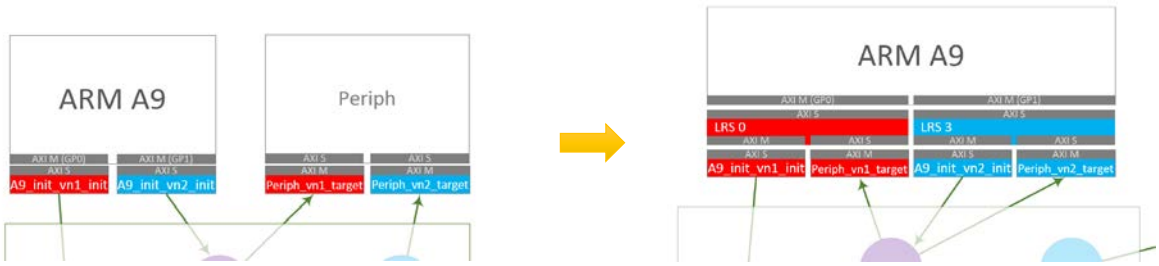

Figure 10 Use of the peripheral target NIs for the incoming traffic at the ARM (left figure represents the initial scenario, without LRS and the right figure represents after the insertion of the LRS)

4.2 WP2

Technologies developed in the WP2 mainly consist of the building blocks of the Dreams Harmonized platform (DHP) for the HW part and Xtratum and KVM for the SW part, to which must be added an intensive modeling work around STNoC for the GEM5 demonstrator. The DHP mainly consists of the STNoC in mixed criticality context enhanced at its boundaries with the LRS HW blocks. to reach outer network node, the DH is provided with a TTETHERNET bridge.

The different technologies developed in WP2 are used in the different demonstrators from WP6, WP7 and WP8. During the intermediate integration phase, several feedbacks have been made by the industrial demonstrator teams. These feedbacks, detailed hereafter, have been addressed by the WP2 in order to support development of the demonstrator.

WP6 – Avionics demonstrator

Avionics Demonstrator will extensively use XtratuM on top of the DHP developed in WP2. On the other side of the network, the avionics demonstrator consists of a T4240 also using XtratuM.

The DHP was pretty well suited for the demonstrator of WP6. WP2 partners are supporting this demonstrator in deploying Xtratum for ARM v7 (the core on the DHP) which is already available and working and in deploying XtratuM for PowerPC which should be available to WP6 demonstrator soon. As well, the Dreams Abstraction Layer is being developed in WP2 to help usage of Xtratum in the framework of WP6 use case.

The communication between the DHP and the T4240 board is achieved by a TTethernet link. Here also WP2 partners are working to integrate TTTech Gateway driver within XtratuM and to enhance communication for these 2 devices.

WP7 – Wind power demonstrator

WP7 demonstrator consists in a Galileo platform coupled to a DHP. For this set up, the communication between the 2 devices is achieved through PCIe (outside of WP2 context). For this use case, the major support from WP2 was to deploy XtratuM to the Galileo platform. This has been achieved and is currently functioning. On the other hand, the DHP as it was first defined was not fulfilling perfectly the needs for the WP7 demonstrator. Particularly the STNoC connectivity and architecture did not make it possible to have 2 microblazes on the highest priority virtual network. Then Its architecture has been revamped to meet these requirements, re-validated and verified and redeployed within the consortium. Currently 2 flavors of the DHP are then available. 1 is used by DP6 and WP8 while the other is used by WP7.

WP8 – Healthcare demonstrator

WP8 demonstrator consists of a DHP connected to a Juno board through the TTethernet gateway. While the developments performed by WP2 partners for supporting the WP6 are extensively reused in this demonstrator, WP8 is the only demonstrator exhibiting the developments performed on the KVM hypervisor for ARM v8 architectures (juno board). As a result WP8 demonstrator development is carefully supported by VOSYS and TEI providing respectively the KVM hypervisor and the memguard algorithm to be deployed as an hypervisor service.

4.3 WP3

The goal of the work package is to develop the virtualization technologies, network components and middleware necessary for the implementation of distributed networking services that support the mixed-criticality requirements from the DREAMS architectural concept. These activities range from the actual communication subsystem (i.e. switches, end-systems, firmware, drivers and configuration) as well as the mechanisms for global reconfiguration and end-to-end security.The support provided

by WP3 is focused on these aspects and the usage and integration of the technologies in the demonstrators as described in the chapters 3.7 (Resource Management), 3.9 (Security Services) and 3.10 (Gateways) of this document.

The main focus of the activities in the partner support during and after the intermediate integration phase consists of providing support to the correct integration of these technologies into the respective higher layer architectures, represented by on the one hand the hypervisors XtratuM and KVM for the aerospace and the healthcare demonstrator, as well as practical support on the usage of the configuration tool chain to generate working and correct configuration for the gateways for the respective scenarios.

4.4 WP4

The goal of this work package is to provide an integrated tool support for a model driven development process of mixed criticality real-time systems. Some of the tools implement sophisticated design and verification algorithms that are developed or enhanced in the context of the project. The algorithms and tools cover three main aspects, which we are going to summarize in the sequel.

A product to be developed often foresees a wide range of combinable optional and mandatory features, which result in a large set of configurations (product line) which are impossible to validate individually. Product sampling and the clean separation between business and technical **variability** are the approaches developed in T4.3 to allow an efficient design and verification of product lines.

Once the business and technical variability are bound, it becomes possible to design the allocation of the execution and communication resources of platform to the applications, while ensuring the execution of the critical application even in the case of the failure, for e.g. a failure of a processor. This is enabled by **offline adaptation strategies** for the scheduling of **mixed criticality** applications developed or enhanced in T4.1.

Last but not least, once a valid resource allocation configuration has been found that satisfies all requirements, it must be made sure that the configuration is translated without alteration in to actual configuration of platform building blocks to ensure the foreseen behavior at run-time. To solve this problem, the **model driven generation of configuration files** is implemented by T4.2

The above mentioned algorithms are often based on concepts that are difficult to grasp for non-specialists. This is a hurdle for their concrete application in the demonstrators, even if tools are available. Therefore, in order to promote the actual application of the algorithms and tools developed in WP5 and to ensure this way the effective applicability of the results, the **demonstrator support** task T4.4 provides an inventory of all tool functionalities and tool chaining possibilities in demonstrator relevant use cases, while assisting the demonstrator partners in identifying opportunities for applying the tools and in actually applying the tools and interpreting correctly their results.

4.5 WP5

4.5.1 Overview

The goal of the work package is to pave the way towards a competitive development and certification of mixed-criticality solutions. Competitive development and certification emphasizes the need for solutions to manage complexity, increase re-usability, reduce product cost and reduce product overall certification cost and time. For this purpose, modular safety-cases, patterns, tool integration, test beds and guidelines are provided based on the architectural style, virtualization, multicore and mixed-criticality network contributions already provided in other WPs. IEC-61508 is considered to be the reference safety standard for this WP.

The work package results are on the one hand certification methods and on the other hand, a simulation and fault injection framework that enables assessment in the demonstrators. Taking into

consideration these results, demonstrator support includes not only help in using the provided tools but also support on the identification of opportunities to apply them. The following activities have been or are going to be performed in the task:

- Define all the documentation that shall be generated and required in a certification process, the tool usage shall be defined too.
- Support to confirm that the process of the development of demonstrator is correct.
- Support the usage of the simulation and fault injection framework, it will be defined a guideline about how to use the framework according to the safety requirements of the demonstrators.

In the following, the demonstrator support by means of the different tasks and results that are expected from the WP will be explained. WP5 results are mainly referred to WP7 wind power demonstrator where all the results are being applied.

4.5.2 Modular Safety Cases

Each modular safety case describes arguments to demonstrate that safety properties are satisfied and risk has been mitigated. Modular safety cases have not been directly applied in all the demonstrators. The demonstrator where the modular safety cases have been applied is the wind power demonstrator developed in WP7. In order to support this demonstrator, modular safety cases have been integrated in WP4 defined toolchain. The integration process has been supported by this WP and the application of the tooling related to the certification process.

4.5.3 Simulation, verification and fault-injection framework

The simulation, verification and fault injection framework is supporting all the demonstrators. These tools have been used to simulate the building blocks of the DREAMS architecture for the cluster and chip level. Simulation tools have supported chip level protocol verification and the investigation of safety properties, assuring that the timing performance of chip level building blocks supports the required behavior of the demonstrators. The fault injection components, for the injection of operational faults and design faults in the simulation components, allow checking if the safety function of the corresponding demonstrator is working properly. This fault injection framework will be used by the wind power WP7 demonstrator when the EtherCAT communication is ready.

4.5.4 Cross-Domain Mixed-Criticality Patterns

Cross-domain patterns are widely used for describing and documenting recurring solutions for design problems of systems, sub-systems or elements. These patterns are used 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). In this case we are creating different patterns according to the different building blocks that we can find inside the DREAMS architecture. The application of some of these patterns is being done inside the wind power demonstrator (WP7). From WP5 we are supporting the demonstrator development by easing the integration of the selected patterns inside the application.

4.5.5 Tool integration in industrial (safety) engineering process

Some of the results of WP5 will be used from a methodological point of view when designing the system. To design the demonstrators, based on the DREAMS architecture, the defined toolset from WP4 is used by integrating results from WP5. Therefore support to the demonstrators through the application of the tool chain is continuous. In the first stage we have supported the integration of WP5 results inside the tools developed by WP4, in this second stage we are supporting the use of these tools by all the demonstrators.

5 Bibliography

[1] Xilinx, ZC706 Evaluation Board for the Zynq-7000 XC7Z045 All Programmable SoC - User Guide, http://www.xilinx.com/support/documentation/boards_and_kits/zc706/ug954-zc706-eval-board-xc7z045-ap-soc.pdf

[2] DREAMS. D1.1.1 - Architecture Conceptualization: Requirements, Terms and Principles 03/2014.

[3] DREAMS. D1.2.1 - Architectural Style of DREAMS, 07/2014.

[4] DREAMS. D1.3.1 - Description of Development Process with Model Transformations, 07/2014.

[5] DREAMS. D1.4.1 - Meta-models for Application and Platform, 03/2015.

[6] DREAMS. D1.5.1 - Intermediate integration of DREAMS platform with virtual platform prototype, 4/2015.

[7] DREAMS. D1.6.1 - Meta-models for platform-specific modelling, 05/2016.

[8] DREAMS. D4.3.2 - First implementation and improvement of variability analysis and testing techniques for mixed critical systems, 11/2015.

[9] DREAMS. D4.3.3 – Final implementation and improvement of variability analysis and testing techniques for mixed critical systems, 07/2016.

[10] DREAMS. D6.3.1 - Preliminary report on DREAMS technologies assessment for avionics.

[11] DREAMS. D7.3.1 - Preliminary assessment report related to improving or calibrating the technological results 05/2016.

[12] DREAMS. D8.3.1 - Preliminary assessment report related to improving or calibrating the technological results 05/2016.

[13] R. Obermaisser. Time-Triggered Communication. CRC Press, Oct 19, 2011.

[14] DREAMS. D4.1.3 – Final implementation and improvement of the offline adaptation strategies for mixed criticality, planned for 07/2016

[15] DREAMSD 4.2.2 – Final implementation of a platform configuration files generator