SELENE: Self-Monitored Dependable Platform for High-Performance Safety-Critical Systems

Carles Hernàndez*, Jose Flich*, Roberto Paredes*, Charles-Alexis Lefebvre†, Imanol Allende †, Jaume Abella ‡, David Trilla‡, Martin Matschnig§, Bernhard Fischer §, Konrad Schwarz¶, Jan Kiszka¶, Martin Rönnbäck‖, Johan Klockars‖, Nicholas Mc Guire**, Franz Rammerstorfer††, Christian Schwarzl ††, Franck Wartel‡‡, Dierk Lüdemann[x], Mikel Labayen[xi]

* Universitat Politècnica de València (Spain)
† Ikerlan (Spain)
‡ Barcelona Supercomputing Center (Spain)
§ Siemens Corporate Technology (Austria)
¶ Siemens AG (Germany)
‖ Cobham Gaisler (Sweeden)
** OpenTech EDV Research (Austria)
†† Virtual Vehicle Research (Austria)
‡‡ Airbus Defence and Space (France)
[x] Airbus Defence and Space (Germany)
[xi] CAFsignalling (Spain)

***Abstract*—Existing HW/SW platforms for safety-critical systems suffer from limited performance and/or from lack of flexibility due to building on specific proprietary components. This jeopardizes their wide deployment across domains. While some research has been done to overcome these limitations, they have had limited success owing to missing flexibility and extensibility. Flexibility and extensibility are the cornerstones of industry adoption: industries dealing in capital goods need technologies on which they can rely on during decades (e.g. avionics, space, automotive).**

SELENE aims at covering this gap by proposing a new family of safety-critical computing platforms, which builds upon open source components such as the RISC-V instruction set architecture, GNU/Linux, and the Jailhouse hypervisor. SELENE will develop an advanced computing platform that is able to: (1) adapt the system to the specific requirements of different application domains, to changing environmental conditions, and to internal conditions of the system itself; (2) allow the integration of applications of different criticalities and performance demands in the same platform, guaranteeing functional and temporal isolation properties; (3) achieve flexible diverse redundancy by exploiting the inherent redundant capabilities of the multicore; and (4) efficiently execute compute-intensive applications by means of specific accelerators.

***Index Terms*—safety, high-performance computing, autonomous systems**

I. Introduction

In recent years, the safety critical systems market has been growing quickly, chiefly because of the advent of new applications such as autonomous driving systems. Autonomy has reached the safety critical systems market spurred by the availability of low cost, reduced size and low power, high-performance embedded computing platforms able to host computationally intensive applications in real-time. However, the validation of safe and reliable operation of such complex platforms (at reasonable cost) is still an open problem that calls for novel and viable design approaches.

SELENE aims at covering this gap by proposing a new family of safety-critical computing platforms enabling the use of Artificial Intelligence (AI) and building upon open source RISC-V-based components. SELENE will develop a high-performance computing platform that is able to: (1) adapt the system to the specific requirements of different application domains, to the changing environmental conditions and to the internal conditions of the system itself; (2) allow the integration of applications of different criticalities and performance demands in the same platform, while guaranteeing functional and temporal isolation properties; (3) achieve flexible diverse redundancy by exploiting the inherent redundant capabilities of multicore architectures; and (4) efficiently execute compute-intensive applications by means of specific accelerators. These features will be achieved while complying with safety requirements and considering security aspects by construction.

To efficiently execute AI on our hardware platform, we will incorporate state-of-the-art machine learning accelerators tailored to safety-critical system applications. AI software will be executed using an execution strategy that enables the coexistence of tasks with different criticalities. In particular, SELENE, using an open source software stack, will allow the coexistence of applications with mixed integrity levels within the same hardware platform preserving their assurance as needed, in line with appropriate standards and qualification guidelines, but without comprising maximum performance that other applications with less stringent safety requirements may require. This will be achieved by using lightweight hardware and software isolation features where needed.

SELENE targets a wide range of safety-critical domains and will provide the community with a key building block to facilitate the development of new autonomous and critical applications in emerging domains. To validate the flexibility of the platform, SELENE will deploy four different use-cases: the SPIDER autonomous robotic platform from Virtual Vehicles, a highly-integrated satellite application from Airbus Defense and Space (France), a human flight space application from Airbus Defense and Space (Germany), and an autonomous train application from CAF signalling. The SELENE HW platform will be prototyped in an FPGA where the complete

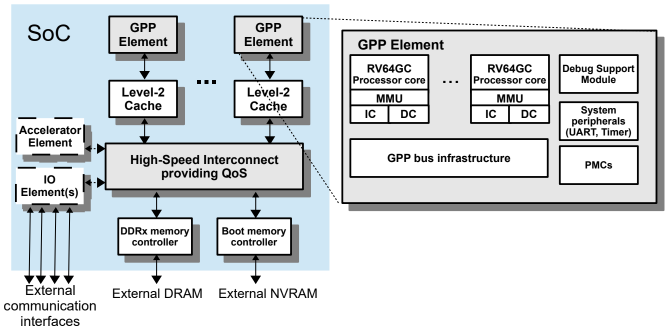

Fig. 1. Baseline SELENE SoC architecture.

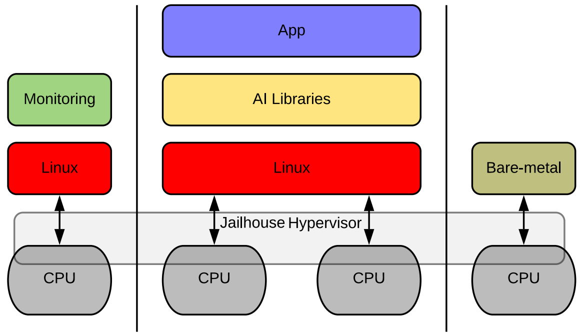

Fig. 2. SELENE SW Architecture.

SW stack will be deployed to support the different use case needs. SELENE will deliver proofs-of-concept at TRL4-5 at the end of the project and will explicitly pave the road towards higher TRL. SELENE started in December 2019 and will last until December 2022.

II. SELENE Hardware Platform

SELENE will develop a heterogeneous multicore processor platform based on the open RISC-V Instruction Set Architecture (ISA). This multicore processor will expose the parallel, redundant computational resources in the chip to meet different safety, performance, security and power requirements for different applications and domains. To allow meeting safety requirements in several domains, SELENE will develop a multi-standard safety-IP library of extended functionalities for testing, diagnosing, debugging and monitoring the platform as well as isolating and protecting the applications executing on it. Mechanisms such as hardware virtualization will be used to isolate applications from each other and protect the system.

The SELENE hardware platform relies on a modular, existing system-on-chip (SoC) design that will be adapted to fit the needs of the project use-cases. Figure 1 shows the architecture of the baseline SoC. The General Purpose Processing (GPP) elements consist of one or several Cobham Gaisler NOEL-V [8] superscalar, in-order RISC-V RV64GC processors. These cores are able to provide the same type of fault-tolerance features as current European space-grade LEON processors meeting the stringent requirements of space end users. The NOEL-V core matches ARMs Cortex-A53 processor performance per MHz. The development work anticipates a future certification effort toward safety standards such as IEC 61508 Ed 2.

Current state-of-the-art microprocessor implementations with extensive sharing of resources suffer from timing interference. The SELENE platform will provide a high-speed interconnect with traffic flow isolation properties to enable guarantees between the contenders for shared resources (GPP, IO and accelerator elements compete for access to non-local memory). A concern for the implementation of safety-critical systems is the observability and guarantee of correct hardware functions on which software depends to provide safety. Thus, the SELENE hardware platform will provide the necessary debug and monitor visibility within and around the SoC elements to meet these needs.

III. SELENE Software Platform

SELENE will exploit the capabilities of a heterogeneous, high-performance, and certifiable hardware platform with the development of an appropriate high-integrity-level-capable open-source software layer that relying on built-in hardware features delivers safety-critical amenable services. For this purpose, SELENE will make direct use of Linux in cooperation with the SIEMENS Jailhouse open source hypervisor running on top of a RISC-V platform. Figure 2 shows the software architecture of SELENE.

A. *Linux for Safety-critical Applications*

GNU/Linux is one of the most popular Operating Systems for non-safety critical systems. Currently, it is the leading OS for supercomputing, public cloud, and smartphone domains. In addition, it has an important role within embedded systems, with an estimated deployment rate of 62% [4]. Although these applications are non-safety critical, different industrial and economical sectors also rely on GNU/Linux for their critical applications, such as the telecommunications industry.

The need for high-complexity applications in safety-critical systems makes Linux a good open source candidate within SELENE. Linux based systems provide a number of mechanisms that allow an optimal utilization of the given resources while providing outstanding security capabilities (protection and monitoring). In order to build a mixed-critical system, it is necessary to establish and maintain sufficient isolation between zones of different criticality, sometimes also between different functionalities with identical criticality (safety function from monitoring, diverse safety function implementations from each other, incremental integration and qualification). When porting and consolidating these functionalities on a system with a shared processor and other shared resources, isolation has to be achieved at a higher level, using a combination of hardware and software measures. One approach is to use the isolation of Linux between applications it can host. Such an approach is taken in the SIL2LinuxMP [14] project and its successor ELISA [6] by using Linux kernel mechanisms to achieve the required separation. Those mechanisms are also used to isolate processes or groups of processes, including containers, from each other and are primarily implemented in software at the operating system level.

However, the number of interfaces and functionalities of the OS kernel that need to be considered to achieve isolation can be significant. They may grow in number, complexity and diversity as the OS evolves limiting the flexibility of the platform. Changes to the OS kernel, e.g. to add functionality for non-critical functions or address security issues in the latter, require re-certification of the OS. Thus, as a complementary solution to such a monolithic approach, SELENE will use an hypervisor that intercepts the interaction of the operating system with the hardware allowing partitioned access to shared resources and potential multiple guest operating systems execution. Hypervisor takes over the burden of isolating workloads from each other that are run in different instances of operating systems, or even with only a minimal runtime environment (bare metal).

B. The Jailhouse Approach to Static Partitioning

The point of introducing hypervisor technology into safety-critical systems is to reduce the burden of certification by "walling in" subsystems. Properly walled-in subsystems cannot impact other subsystems. Hence, only the safety-critical subsystems and the hypervisor itself require certification.

However, depending on its functionality, the complexity of a general-purpose hypervisor can easily match or exceed the complexity of an operating system. In that case, moving the partitioning duty from an OS to a hypervisor would not simplify system certification. To the contrary, it would make it more complex by effectively duplicating the effort needed for system-level software (OS used for safety critical parts and hypervisor).

The central tenet underlying the design of Jailhouse is to ease certification through maximally simple design. Jailhouse was developed around a concept of static partitions used to keep the independent execution of operating systems in check.

Jailhouse key design principles are:

- No scheduling, only 1:1 resource assignments:
  - One (or more) CPU cores $\mapsto$ one guest. Multiple guests are never scheduled on the same CPU core.
  - One device $\mapsto$ one guest. Device sharing is managed by the guest owning the device (by providing services to other guests).
- Focus on maintaining static partitions: dynamic reconfiguration while the critical system parts are in service is not possible.
- Freezing platform hardware configuration after boot-up in known-good state (late partitioning), rather than booting the system via the hypervisor and virtualizing the boot-up of standard guests. (This depends somewhat on the processor architecture).
- Prefer matured hardware-assisted virtualization and resource partitioning features over software approaches (including paravirtualization)

Following these principles, the current code base of the Jailhouse hypervisor is able to stay below 10,000 lines of code (C and assembly) per target architecture (currently supporting x86, ARMv7 and ARMv8). This offers a high degree of flexibility with respect to certification approaches and the safety-conforming management of the code base. These are the main reasons for its deployment and extension to support additional safety and security-related features in the SELENE computing platform.

C. Artificial Intelligence Libraries

A key aspect of autonomous systems is the use of AI techniques including considering these advanced technologies for critical decision making. This only can be tackled in a safe way if the highly complex AI components can safety co-exist with low-complexity checkers or monitors as well as diverse independent concurrent implementation forming architectural protection schemes for complex technologies.

We will deploy artificial intelligence (AI) techniques on the SELENE platform isolating these by partitioning mechanisms provided by Jailhouse and GNU/Linux to enforce non-interference with other critical components. In particular, in the context of the SELENE project we will port the European Distributed Deep Learning Library [15] to the SELENE computing platform.

Common AI methods currently are not assessed for robustness against random faults, notably it is unclear at present what inherent robustness properties they may possess against control-flow errors or data errors. Given the algorithmic robustness against input noise it can be speculated that AI algorithms actually fare well with some types of random faults in data - what level of robustness they actually expose is though an open question. Further the issue of protecting against systematic faults in a highly complex pre-existing element like EDDL is an open issue. How to protect an AI executable so as to allow giving guarantees - or at least a quantitative assurance of the results emitted is a topic we will investigate in the context of SELENE deployment of EDDL.

Safety related executables making critical decisions based on AI require AI systems to provide a quantification of their uncertainty (algorithmic as well as their susceptibility to random faults). A research target of SELENE is to explore options to quantify the reliance that can be placed on AI components in a safety related system.

Methods to achieve protection of AI components seen as worth exploring are diversity based approaches currently being used in ensemble based optimizations. Such methods may also serve the safety needs for fail-operational autonomous systems. In SELENE we propose to use ensembles of neural networks in order to provide robustness of model rather than focusing on optimization issues. The challenges raised are the considerable memory needs as well as the computational demands of AI (e.g. deep convolution neural networks) which run contrary to the traditional keep-it-simple principle. Questions of how to achieve adequate system level integrity for the class of RAM/CPU intensive tasks, in general and considering randomness based algorithms commonly used in AI, are currently not addressed by FuSa standards and state-of-the-art FuSa techniques.

IV. Flexible Self-Monitored Platform for Safety Critical Applications

The goal of SELENE is to build a computing platform that facilitates the certification of complex safety critical applications. To achieve this goal, SELENE relies on two key aspects: on-line monitoring capabilities, and spatial isolation.

SELENE monitoring capabilities will be provided with specific hardware support that will be triggered by software means. In particular, hardware **monitoring support** will be incorporated in those components whose behaviour can lead to safety goal violations. These critical components are for instance the memory management unit (MMU), the performance monitoring unit (PMU), and the various fault detection and correction mechanisms (e.g error correction codes, and comparators). To enable software-triggered monitoring the SELENE platform includes a hardware monitoring unit (HMU). The goal of SELENE monitoring capabilities is twofold. First, during the verification step, the HMU allows collecting evidence on conforming to safety requirements. Once the system has been deployed, the HMU allows software monitors and applications to learn about the actual status of the platform during the lifetime of the same; raising alarms when limits have been exceeded.

Spatial isolation will be applied by leveraging the capabilities of the Jailhouse hypervisor. Additional hardware support combining isolation and monitoring capabilities will be included in these hardware features for which software virtualization is not enough (see Figure 3). For instance, shared resources such as caches and the network-on-chip (NoC) will be adapted in SELENE to enable the co-existence of tasks with different criticality in the same hardware platform. SELENE interconnect will be designed to combine innovative techniques for partitioning and monitoring [3], [13]. Physical NoC partitioning will be carried out at a coarse-grain level using virtual channels or virtual networks to isolate different traffic flows. We will also explore more fine-grain isolation techniques using TDM-based packets scheduling [13].

Finally, SELENE will support multiple configuration options (**flexibility**) to address safety and performance needs from diverse applications. Limited but key modifications in specific hardware components allow reconciliation of requirements from several application domains on a single computing platform. Figure 3 shows a particular instance of the SELENE platform. This plot shows how the SELENE platform has been adapted to execute two highly critical fail-operation tasks and a performance demanding AI application.

A. *Flexible Support for Diverse Redundancy*

Safety-critical systems are designed to be robust against hardware faults. This is specially important in some environments (e.g. space) where random hardware faults can be more frequent and potentially lead to system failure. To shield such systems against that possibility, engineers introduce mechanisms to detect and mitigate faults that can affect the entire platform and compromise its safety.

Especially for complex processor cores targeting safety critical applications proper verification and fault diagnosis throughout the whole design process are essential. SELENE sets special focus on fault analysis techniques to enable early detection and mitigation of common-cause faults. One major research area is about closing fault coverage holes by proposing additional safety mechanisms and estimating diagnostic coverage. Important questions in product design adress which safety architectures are best suited with respect to power, performance and resource utilization (e.g. area). Safety challenges are approached by design flow enhancements such as safety analysis and automated insertion of safety measures followed by fault campaigning to validate the architecture together with the safety mechanisms.

For instance, to avoid Common Cause Failures (CCFs), critical systems can replicate components and slightly change some of their characteristics, so even though they are identical in functionality, a fault affecting the replicated components will manifest in different ways, therefore enabling its detection and preventing a fault from becoming a failure. In that regard, most commercial processors [7], [10] implement a Dual Core LockStep (DCLS) which consists in replicating two identical cores executing identical software but staggering their execution some clock cycles. In that way, if a random fault occurs at the same time in both cores, it will affect two different processor states and when outputs are compared the fault will be detected.

DLCS comes at the expense of more area and power and sacrifice the computational performance of at least the replicated cores, since those are entirely dedicated to replicate the execution of a task. Under current performance demands, devoting cores to code replication might imply that the most performance demanding applications cannot be executed properly or will suffer from lack of resources.

In that regard, the SELENE platform aims at providing flexible lockstep, in which critical tasks will have the capability of executing in safe mode using ”locksteped” processors while the more performance demanding and less critical tasks can switch to a high-performance mode where the cores can execute distinct software freely. This will provide flexibility to the SELENE platform and enable its use for the upcoming applications (e.g. AI). In Figure 3 Hardware Monitoring Unit (HMU) modules are added and connected to the cores to enforce lockstep or high-performance mode. This modules monitor the activity of the cores when needed and mandate resets or stalls in safety mode.

In safety-critical real-time systems, tasks are allocated a time budget derived from Worst-Case Execution Time (WCET) estimates. System integrators must, therefore, provide evidence that deadlines are met and tasks adhere to their time budgets. Typically, to obtain such evidence, the timing Validation and Verification (V&V) process consisted in extensive testing campaigns that demonstrated that no overruns occurred.

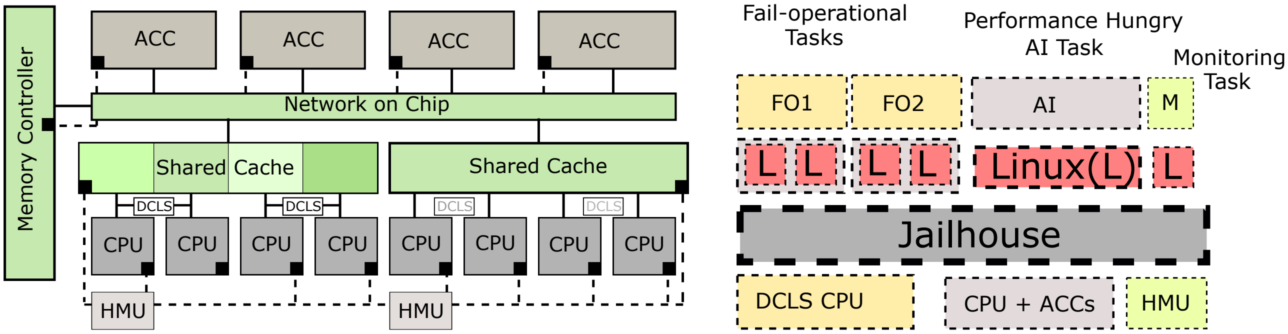

Fig. 3. Particular instance of SELENE platform. Hardware view (left) and logical view (right).

Up until now, end-to-end measurements were enough to provide evidence of correctness and adherence to the timing budgets. This was possible because critical hardware platforms were sufficiently simple so that users could master them and create specific platform tests that exercised the worst-case scenarios. However, with the advent of Artificial Intelligence (AI) and new applications, an increase in computational performance and complexity is changing the architecture of critical systems (e.g. by introducing multicores, accelerators) such that typical forms of timing V&V fall short of providing the evidence required.

In this context, multicores are the cornerstone of performance improvement in real-time systems. The key concept is to integrate the performance of several processors at a reduced cost by sharing specific resources. Unfortunately, resource sharing impacts the timing behavior of tasks that originally executed in isolation and now have to compete to access certain processor features. Resource sharing also affects timing analysis as different tasks exercise different interference, therefore, compromising the ability to further integrate the system. In that regard, end-to-end measurements no longer provide the detailed information required to integrate tasks in a multicore platform. For example, anomalous resource contention can be masked when using end-to-end measurements which do not provide a break-down in resource contention, thus compromising the V&V and integration effort. Moreover, the end-to-end approach lacks the information needed to diagnose during operation time anomalous behavior like deadline overruns. To address such problem, finer grain solutions involving the use of Performance Monitoring Units (PMUs) have been proposed to acknowledge where interference is coming from [5].

Overcoming this challenge is a key aspect of providing the performance needed in safety-critical platforms. In that regard, the SELENE platform will provide support for multicore timing V&V through the implementation of an extended PMU that will provide fine-grain information allowing the user to differentiate the sources of timing interference and act accordingly. SELENE's PMU will enable the user to reason on the amount of interference that each shared resource exercises on the final execution time and which agent is utilizing that resource. Figure 3 depicts the use of SELENE's HMUs in a distributed architecture. Note that PMUs are key component of this larger monitoring module.

B. Managing complexity to certify software-intensive complex-systems

There is a preference in industry to manage safety related elements independent of their context. A recent materialization of this notion is SEooC (Safety Element out of Context) in ISO 26262 Ed 2 Part 10. Furthermore, traditional OS or middleware elements (e.g. network stacks) have been certified independent of a specific use-case and then used as ”drop-in” safe-OS. Nevertheless, looking at the history of those pre-certified OS elements, it can be concluded that they were not pre-certified from the start but rather, after being deployed in multiple contexts, a common base emerged and was then subjected to certification. From the SELENE perspective, the relevant question is, therefore, whether this could be immediately applied to GNU/Linux after 10 years of experience using it in some safety-critical systems [9]. As explained in the following paragraphs, we foresee this as not doable due to the significant increase of the overall complexity of the system software.

IEC 61508 defines 2 basic types of elements: Type-A (low complexity) and Type-B (high-complexity or formally simply non-low complexity). Type-A is roughly defined as systems where (1) all failure modes are known, (2) behavior under fault condition is understood and (3) adequate data is available to support the claims. Type-B is a system where one of these three conditions is not satisfied. Whereas the certification of the traditional OS out-of-context mentioned before was based on the accomplishment of Type-A conditions, in highly complex systems targeted by SELENE clearly all three conditions for Type-A are violated.

Not only that, but the past certification of GNU/Linux based safety-critical systems was only possible due to the accomplishment of the following assumptions: 1) simplicity of safety related software systems and 2) total ignorance of security aspects as well as 3) assumption of static configuration over the life-time of systems (or very infrequent updates). Now again, in the anticipated highly complex autonomous systems targeted by SELENE, these assumptions will necessarily be violated.

In order to ensure that these software-intensive complex systems can provide the required services with a tolerable level of risk while considering economic constraints, the following key aspects of system engineering need to be addressed: (1) minimizing the design, (2) managing dynamics of change, and (3) joining safety and security. Finally we see potentials in actually reversing the approach and anticipate capitalizing on complexity rather than trying to keep systems simple.

Design minimization. Based on the preliminary conclusions of SIL2LinuxMP project [14] SELENE proposal is to implement this paradigm by maximizing the context awareness when selecting and configuring elements: functional, architectural, safety, security context. In this way, it is possible to know what part of an element is intended for use under what specific constraints and then only is the impact of deviation from design intent understandable. For example, if we know that memory allocation is done only before any safety related activities commence, then we can either assess the potential impact of memory allocation failures as negligible (unlikely) or mitigate them by use-case specific measures that only cover those specific aspects of the memory allocations functional variety that pertain to hazardous consequences.

In other words, managing complexity by maximizing contextual constraints will be the starting point for SELENE. Much like traditional RTOS and generic safety related components, we first must gain adequate understanding under severely limiting conditions and then expand from there on, by extraction of commonalities, to a (partially) context independent verified/trusted subset at the design level.

Managing dynamics of change, and joining safety and security. At the code level limiting dynamics will simply never be achievable due to the change rates, so we need other strategies to manage dynamics. These will mandate the ability to perform impact analysis in a very short time (24-48h) allowing to respond to critical situations as well as an update strategy that allows to temporarily impose limitations on systems (for the initial analysis respectively fix-rollout time) followed by a seamless continuation of operations after a transparent update. Clearly this has significant impact on the methodology and procedural aspects of certification and the overall socio-technical systems life-time.

Capitalize on complexity. One of the most significant changes that safety related autonomous systems are undergoing is the increase in system complexity. While most will agree to that, the definition of complexity (e.g. as outlined by Melany Mitchell in [11]) and the impact of the same on safety is not so clear. For the later guiding standards like IEC 61508 essentially strive to avoid it as the root of (almost) all evil expressed well in its association of complexity with limited ability to understand [1]

Complexity is commonly associated with emerging properties - that is properties that are not present in the individual elements comprising the overall system, but rather properties that only manifest themselves in the interaction of the connected elements (via intended and unintended interfaces). The direct result is that for complex safety related systems the previous divide and conquer strategies painfully fail and assurance is bound to our ability to analyze the system as a whole. Nevertheless we as humans have quite limited abilities to analyze complex systems without the guidance by ”law of parsimony” in its different guises. To regain the ability to manage highly complex systems analytically we need an orthogonal approach to ”divide and conquer” and this approach may be ”contextualization” - with other words boil down the context to an absolute minimum so as to allow expenditure on the increased behavioral interdependence.

At the same time, we need a change of strategic focus. Complexity shall no longer be viewed solely as an unfortunate and unavoidable byproduct of advanced systems but needs to be viewed as an opportunity to mitigate failures too. How could this be achieved ? For instance, systematic faults introduced not by the faulty functionality of the individual element but by their interaction with other elements can profit from the non-deterministic behaviour of complex systems. In other words - viewing a function as a black-box in a complex system - if a desired property implementation results in a deterministic interaction then traditional analysis suffices, else the non-reproducibility as a palpable manifestation of complexity implies that concurrent executions would potentially divert and thus be diverse to some extent. Arguments building on such inherent properties of complex systems are the can-of-worms they present us with to extend our safety functional mitigation toolbox.

Identifying the potentials of complex systems - loosely coupled lock-step inherent diversity [12] or logical isolation [1] as well as a number of non-deterministic methods employed in security need to be added to our safety tool-box. This changed view on complexity may allow us to address common cause faults or mitigate unavoidable undesired interference - to single out just two - in complex systems, so that we can capitalize on complexity rather than continue the long lost battle for perceived simplicity. One of the few things that seems sure is that complexity in safety related systems is here to stay.

V. Use-cases

A. Human Space-Flight

Since the beginning of the Columbus project, Airbus Defence and Space played a key role in the engineering that lead to a unique and successful conjoint history of the International Space Station.

In terms of safety critical and high reliable equipments, the Columbus flight module infrastructure subsystems present many examples. The major subsystems, namely Thermal Control System (TCS), Environmental Control/Life Support Subsystem (ECLSS), Data Management System (DMS) and Electrical Power Distribution System (EPDS) constitute the safety critical infrastructure elements which make up the

[1] IEC 24765 3.500 complexity: *1. the degree to which a system’s design or code is difficult to understand because of numerous components or relationships among components 2. pertaining to any of a set of structure-based metrics that measure the attribute in (1). 3. the degree to which a system or component has a design or implementation that is difficult to understand and verify.*

Columbus Module basic functionality. They provide and control the environment for the astronauts to live in, as well as the cooling, electrical and data interfaces for the experiments which are executed inside the Columbus module.

Airbus continuously considers portability and replaceability of subsystems due to component deprecation. SELENE will help to build the foundation of future hardware and software solutions of the Columbus module.

SELENE will also impact future space missions of mankind. For instance the space Gateway is a new intended space station in the lunar orbit. It is planed to be built within the next upcoming years and will fulfill two main objectives. It enables humans to go back to the moon and it is the first step for a mission to Mars.

Human space-flight relies more and more on deep integrated support by robotic equipment to increase autonomy and allows the astronauts to focus on their essential work. One example of robotic arms under investigation by AirbusDS is the CAESAR [2] of DLR. We expect from SELENE to be the basis for more demanding robotic arm use-cases, for instance in the context of in-space assembly [16]. Within the scope of SELENE, we expect following tasks:

- The multi-core platform of SELENE will allows us test the execution mixed critically application of the robotic arm, separating for instance the telemetry tasks from the robotic control unit.
- We will benchmark if SELENE is capable to increase the computational power for demanding operations in comparison to previous used space grade equipment.
- Especially, we intend to explore the provided AI-acceleration of SELENE to check whether it meets future requirements of the image processing that is essential for the operation and includes a huge safety impact.

B. Very Integrated Satellite Architecture

State of the art data management and control in Satellite platform systems is currently implemented on single processors and typically composed of the Attitude and Orbit Control system, the housekeeping and management functions at satellite, platform and payload levels including Failure Detection Isolation and Recovery mechanisms, and the Data Management System. The global criticality level is usually homogeneous among all those software components, which are validated and qualified as a whole monolithic piece of software.

With the increase of processing power and the push for reducing costs, weight, volume and power consumption, integrated and modular architectures have been recently deployed, thanks to partitioned execution platform based on hypervisors. This enabled the integration on a single computer of the aforementioned Central Software functions together with the more demanding data processing functions such as GNSS and Star Tracker applications which were historically implemented in separate computing devices.

In future systems, with more CPU performance available on board and more use of mainstream standards from other industrial and commercial domains, the processing power could become sufficient to integrate almost all computing functions on a single device for many types of spacecrafts, in particular also the payload control and data processing functions. SELENE supports such opportunity by supporting mixed criticality and providing the necessary safety and dependability assurance.

Within the SELENE project, the satellite use case aims at highlighting and demonstrating the following objective and benefits:

- Simplification of S/W system integration thanks to incremental application development and validation
- Fulfillment of both mixed criticality and high performance requirements
- Simplification of the payload processing design thanks to usage of execution platform with Linux available onboard
- Reduction of cost, weight, volume and power consumption through software and hardware integration

C. Autonomous Robot

The Smart Physical Demonstration and Evaluation Robot (SPIDER) is an autonomous vehicle, with omni-directional movement capabilities, a weight of around 380kg and a top speed of 50km/h, developed by Virtual Vehicle Research GmbH. Given the technical specification, the SPIDER has to fulfill highest requirements regarding safety and security preventing human harm. The technologies covered in this demonstrator in which safe and autonomous decisions are required span from autonomous vehicles, to robotics, and automation industry. In this demonstrator we are deploying the Linux OS on top of multi-core architectures prototyped in FPGAs that are similar to the ones found in commercial boards for autonomous systems in the edge layer.

The SELENE platform will be integrated into the SPIDER to run two safety and security-relevant vehicle functions, namely the collision avoidance function (CoA) and the path tracking function (PTF). In particular, the separation capabilities developed in SELENE are of interest for the SPIDER and self-driving vehicles, because they ensure freedom from interference between the two independent functions, as required for safety-critical functions according to ISO 26262. In addition, computationally intensive applications, like sensor fusion or creation of an occupancy grid are executed on the SELENE platform, in order to verify its applicability for the automotive market.

D. Automatic accurate stopping and safe passenger transfer based on CV&AI-enhanced techniques

CAF Signalling is involved in different research projects related to Computer Vision (CV) and Artificial Intelligence (AI) enhanced systems developed in order to reach a higher autonomy in urban vehicles and align them with railway European normative. The objective is to apply CV&AI techniques to improve different autonomous train operation functionalities such as precision stop, visual odometry, rolling stock coupling operation or person and obstacle detection-identification in railroads.

However, similar to many companies across the sector, CAF Signalling is facing different computational capabilities challenges for CV&AI-enhanced autonomous train operation which needs real-time & safety-critical computing platforms for correct performance. The future of CV&AI breakthroughs in the railway sector will require large arrays of memory devices at the same accuracy as a Graphical Processing Unit (GPU)-based system, hardware accelerators and new platforms. These achievements will expand the scale of CV&AI processing-calculations making them larger and faster (this means energy-efficiency must improve dramatically).

CAF Signalling will use the SELENE approach on CV&AI-enabled computing platforms to execute some functionalities developed in CV&AI field for autonomous train operation. More precisely, SELENEs project use case will focus on:

- **Automatic platform detection** The system will detect the platform area based on train localization information (odometry sensors, balise information . . .) and different visual pattern (visual sensors) detection/identification (characteristic patterns which identifies train platforms). Platform detection functionality will enable CV&AI based automatic train approximation to accurate train stop.
- **Automatic accurate stop at door equipped platforms aligning the vehicle and platform doors** The system will perform precise localisation inside platform area using visual patterns detection, identification and tracking in order to reach an accurate stopping point and managing automatic train operation (traction and brake commands, ATO functionality). The visual patterns will be designed and chosen to maximize the detection and identification processes results in any possible lightness and meteorological conditions. On the other hand, these patterns will be installed according to predefined precise distances to obtain physically accurate measurement from correctly calibrated visual sensors.
- **Safe passenger transfer** The system will manage automatic safe door enabling (ERMTS functionality), making sure the train is completely stopped in the platform area (using visual sensors) avoiding door opening operation if the train and platform doors are not precisely aligned.

VI. Conclusion

In this paper we have described the hardware and software components of the H2020 Self-Monitored Dependable platform for High-Performance Safety-critical Systems (SELENE) project and the applications that will be used to demonstrate its capabilities.

The focus of SELENE is the development of an open platform for high-performance safety-related applications using the RISC-V instruction set architecture. To that end, the SELENE platorm will incorporate built-in support for functional safety in hardware and software to facilitate the certification of complex applications. The SELENE platform is based on the NOEL-V processor and open-source software components like GNU/Linux, the Jailhouse hypervisor, and the European Distributed Deep Learning Library (EDDL).

Acknowledgments

This work has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement no. 871467.

References

[1] Imanol Allende, Nicholas Mc Guire, Jon Perez, Lisandro Gabriel Monsalve, Nerea Uriarte, and Roman Obermaisser. Towards linux for the development of mixed-criticality embedded systems based on multi-core devices. In *2019 15th European Dependable Computing Conference (EDCC)*, pages 47–54. IEEE, 2019.

[2] Alexander Beyer, Gerhard Grunwald, Martin Heumos, Manfred Schedl, Ralph Bayer, Wieland Bertleff, Bernhard Brunner, Robert Burger, Jrg Butterfa, Robin Gruber, Thomas Gumpert, Franz Hacker, Erich Krmer, Maximilian Maier, Sascha Moser, Josef Reill, Maximo Roa, Hans-Jrgen Sedlmayr, Nikolaus Seitz, and Alin Albu-Schffer. Caesar: Space robotics technology for assembly, maintenance, and repair. In *69th International Astronautical Congress (IAC)*, 10 2018.

[3] Jordi Cardona, Carles Hernández, Jaume Abella, and Francisco J. Cazorla. Maximum-contention control unit (MCCU): resource access count and contention time enforcement. In Jürgen Teich and Franco Fummi, editors, *Design, Automation & Test in Europe Conference & Exhibition, DATE 2019, Florence, Italy, March 25-29, 2019*, pages 710–715. IEEE, 2019.

[4] Jonathan Corbet and Greg Kroah-Hartman. Linux Kernel Development Report, 2017.

[5] E. Mezzetti et al. High-integrity performance monitoring units in automotive chips for reliable timing V&V. In *IEEE Micro*, January 2018.

[6] Linux Foundation. *Enabling Linux in Safety Applications*, 2019. https://elisa.tech.

[7] Cobham Gaisler. *LEON3FT fault tolerant processor*, 2019. https://www.gaisler.com/index.php/products/processors/leon3ft.

[8] Cobham Gaisler. *NOEL-V Processor*, 2020. https://www.gaisler.com/index.php/products/processors/noel-v.

[9] Andreas Gerstinger, Heinz Kantz, and Christoph Scherrer. Tas control platform: A platform for safety-critical railway applications. *ERCIM News*, 2008, 2008.

[10] Infineon. *AURIX. 32-bit microcontrollers for automotive and industrial applications. Highly integrated and performance optimized*, 2019. https://www.infineon.com/dgdl/Infineon-TriCore_Family_BR-2018-BC-v03_00-EN.pdf?fileId=5546d4625d5945ed015dc81f47b436c7.

[11] Melanie Mitchell. *Complexity: A Guided Tour*. Oxford University Press, 2009.

[12] William Okelo-Odongo Peter Okech, Nicholas Mc Guire. Inherent diversity in replicated architectures. 10 2015.

[13] Tomás Picornell, José Flich, Carles Hernández, and José Duato. Dcfnoc: A delayed conflict-free time division multiplexing network on chip. In *Proceedings of the 56th Annual Design Automation Conference 2019, DAC 2019, Las Vegas, NV, USA, June 02-06, 2019*, page 95. ACM, 2019.

[14] Andreas Platschek, Nicholas Guire, and Lukas Bulwahn. Certifying linux: Lessons learned in three years of sil2linuxmp. 02 2018.

[15] H2020 DEEPHEALTH project. European distributed deep learning (eddl) library, 2020. https://github.com/deephealthproject/eddl.

[16] Maximo Roa, Korbinian Nottensteiner, Armin Wedler, and Gerhard Grunwald. Robotic technologies for in-space assembly operations. 06 2017.