Multi-Partner Project: UP2DATE4SDV

Enabling Safe and Secure Modular Updates, Upgrades and Dynamic Task-Reallocation and -Execution for Software Defined Vehicles

Gregor Nitsche*, Patrick Uven*, Hannes Fuchs†, Marcus Hähnel‡,
Mijangos Ane§, Enrico Mezzetti¶ and Kim Grüttner*
*German Aerospace Center (DLR), †AVL List GmbH (AVL), ‡Kernkonzept GmbH,
§Ikerlan Technology Research Centre, ¶Barcelona Supercomputing Center (BSC)

***Abstract*—The European automotive industry is undergoing a revolution by the upcoming technologies of software-defined vehicles (SDV) and connected cooperative and automated mobility (CCAM). In a globally challenging context, in which Europe lost market share, the local automotive software and electronics market still expects a 11.9% compound annual growth rate from 2025 to 2030 – even more accentuated with the increasing adoption of advanced driver assistance (ADAS) and autonomous driving (AD). Both SDV and CCAM are key technological paradigms for enabling a shift of the European automotive sector towards a regained strategic competitiveness. Targeting the resulting need for faster development, deployment and test cycles the Horizon Europe RIA UP2DATE4SDV aims to develop a comprehensive ecosystem for seamless and efficient, safe and secure software updates, hardware upgrades, and situation-dependent reconfigurations of SDVs. For that goal, the UP2DATE4SDV consortium collaborates on the definition and development of two abstraction layers – the hardware abstraction layer and the operating system & middleware abstraction layer – as well as on researching and prototyping a safe and secure orchestration and reconfiguration plane between vehicle and cloud. Based on the resulting modular architecture concept and the corresponding DevOps process the project develops demonstrators to showcase safe and secure updates, upgrades and dynamic task reallocation for automotive hardware and software components. In this paper, we introduce the project, its objectives and planned results, draft first outcomes by refining our demonstrator definitions, and conclude with an outlook into the automotive future based on a safe and secure adaptive SDV stack.**

***Index Terms*—Horizon Europe, Automotive, Update, OTA, Upgrade, Reconfiguration, Safe, Secure, Modular, E/E-Architecture**

I. Introduction and Challenges

UP2DATE4SDV is a Horizon Europe RIA started in 2025 and aims to develop a comprehensive ecosystem for seamless and efficient software (SW)-updates, hardware (HW)-upgrades, and situation-dependent reconfigurations of software-defined CCAM vehicles (SD-CCAM-V). The project is running in a challenging context [1], but in an emerging and important sector [2] which requires research and development to shift towards level 4+ automation.

Funded by the European Union under Horizon Europe grant agreement No. 101203060. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.

The automotive industry is transitioning from decentralized to centralized E/E architectures to address increasing complexity, higher computing demands for features like automated driving, and growing cloud integration. Traditionally, this meant each vehicle system, such as engine management or ABS, relied on a dedicated electronic control units (ECUs) communicating independently via bus systems. As each additional ECU required its own HW, wiring and integration, this fragmented approach increased vehicle cost and weight, limited computing power and complicated maintenance and updates. Centralization simplifies development, improves safety and security and enables software-defined vehicle (SDV), fundamentally changing vehicle development, operation, and maintenance [3].

A centralized architecture addresses these challenges by consolidating functions onto high-performance ECUs or domain controllers, enabling control of multiple heterogeneous functions previously managed by specialized units. Combined with virtualization and high-speed networks, this improves data processing and communication. Cloud integration further enhances development through Development and Operation (DevOps), provides continuous monitoring of complex AI functions, adding a safety layer to traditional verification and brings over-the-air (OTA) updates, allowing manufacturers to continuously deliver new functions, safety updates, and bug fixes to a large vehicle fleet. This facilitates continuous SW and HW updates, increasing vehicle value and functionality throughout its lifetime, and is becoming crucial for future viability.

Challenges

At the same time, enabling comprehensive system updates, upgrades and re-configurations on such SDV architectures comes with the following challenges, that are considered in the project:

C1. Composition with heterogeneous local/remote components for a dynamically evolving system: The traditional paradigm of fixed-function ECUs is shifting towards composed, heterogeneous, local (vehicle) and remote (infrastructure, cloud) SW components decoupled from specific HW platforms, enabling evolution over time (see C2 and C3). A similarly heterogeneous composition is taking place at HW level, where different HW components are combined. This requires the specification, design, verification and management of plat-

form independent, connected and updatable SW components. Consequently, this requires appropriate HW combinations to provide sufficient resources. Furthermore, this requires mapping, dynamic deployment, and runtime services for reliable, secure, and updatable operation of SW components across the vehicle-to-cloud continuum (C2, C3, C4). European OEMs are strategically increasing automation levels [4], [5] in combination with more complex operational design domains (ODDs). This requires a system capable of OTA updates and upgrades for automation functions while the vehicle is in service.

C2. Update of any part (HW, SW, application, module, either local/remote): As described in C1, decoupling SW from HW execution platforms enables, and requires at the same time, the independent update of HW and SW components, which both can trigger the need for further HW upgrades. This introduces the challenges of managing new dependencies between HW and SW, maintaining abstraction layers – from the hardware abstraction layer (HAL), over the operating and software system abstraction layer (OAL) to the SW module container interface abstraction – and creating a mapping between SW resource requirements and composable HW capabilities, allowing support for HW-SW decoupling.

C3. Enable synchronized updates for overall system consistency: While C2 referred to in-place updates of individual SW or HW components, big potential of dynamically evolving systems lies in updating complex functions distributed across the edge-cloud continuum, requiring synchronized updates of multiple components. But, this results in challenges for maintaining SW component dependencies, modeling and analyzing overall system dependencies, and implementing an orchestrated update mechanism with integrated consistency checks.

C4. Preserve safety and security in a dynamically evolving system: Preservation of safety and security is of highest priority for the introduction and operation of level 4+ automation in Cooperative, Connected, and Automated Mobility (CCAM) systems, and adoption of new standards will be essential to accommodate emerging technologies. Now, dynamically evolving systems, as described in C1-C3, introduce new vulnerabilities and potential attacks, and distributed updates might create unintended safety and security interactions, that need to be considered.

The following sections explain how UP2DATE4SDV resolves these challenges, introducing the project vision and the objectives in Sec. II, the key results in Sec. III, the basic concepts in Sec. IV, and the demonstrators for evaluation in Sec. V.

II. Vision and Objectives

The UP2DATE4SDV vision is to create a comprehensive ecosystem for updatable, upgradable, and reconfigurable SD-CCAM-V. This will be achieved by developing essential abstraction layers, advanced orchestrators, and an automated robust realization environment, all while adhering to safety and cybersecurity standards. Figure 1 presents an overview of these parts, to be explained by the following objectives and results.

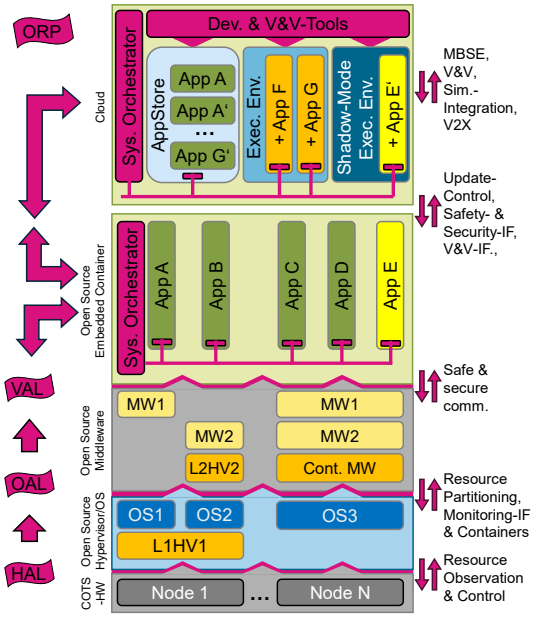

Fig. 1. The UP2DATE4SDV ecosystem for updatable, upgradable, and reconfigurable SD-CCAM-V comprises mainly of the HAL, the OAL, and the ORP together with development, validation- and verification-functions for the update development in the cloud.

To achieve this vision, the project will target the following specific objectives, validated by corresponding key results and associated KPIs.

O1. Creation of a reconfigurable cloud connected zonal vehicle architecture blueprint based on heterogeneous COTS HW (exploiting and building on a unified HAL): In CCAM, system architectures include local computing nodes, in-car connected heterogeneous computing nodes via wired networks, and cloud-connected systems such as cars and urban infrastructures linked through wireless networks. This objective focuses on providing appropriate HAL interfaces for such multilevel architectures. At device level, the HAL will provide means to configure and observe the computing nodes (high-performance commercial off-the-shelf (COTS) devices) and the communication infrastructure based on HW-independent primitives, making SASE properties decidable at the higher abstraction layers building on those primitives.

O2. Establishing a resource composable SW deployment, execution and communication reference layer on top of existing open-source hypervisor and operating system (OS) stacks (exploiting and building on a unified OAL): To support platform-independent development, validation & verification, and deployment, as well as seamless integration, scalability, and maintainability of application SW for the diverse platform configurations and architectures it is necessary to establish a unified SW-abstraction layer on top of the HAL to safely isolate application SW, and to decouple the application layer from dependencies to both HW components of the platforms and their lower-level SW stacks. Based on existing open-source hypervisors, OS and middleware components the common OAL will provide an inter-operable, common interface to the SW application layer on two sides: locally, on each vehicle, and

online, in the cloud, where additional SW can be executed and accessed. This is ensured by using container technologies that can transparently support cloud-based and vehicle-local execution as well as an adaptable communication middleware that facilitates communication inside nodes, across them and into the cloud.

O3. Realization of a dynamic update, orchestration & reconfiguration plane (ORP): This objective aims at realizing a dynamic ORP that allows and ensures safe and secure, synchronized SW OTA updates, HW upgrades and dynamic task-reallocation from vehicle to cloud and from cloud to vehicle. Based on the interfaces of the OAL and the HAL, and the safety- and security-concepts resulting from objective O5, the architecture and the processes of the ORP will be defined. Furthermore, the integration of local and remote verification services for integration testing will be considered according to objective O4.

O4. Provide harmonized and simplified development, integration and validation processes for SDV applications: To achieve faster SDV application development, while ensuring quality and compliance with safety and security standards, the project will provide a harmonized and automated validation and verification framework relying on virtualization of the upgradable UP2DATE4SDV in-vehicle electronic control architecture and utilizing a model-based systems engineering (MBSE) methodology and toolchain tailored for next-gen AI-based and sensor-fused automated and connected vehicle applications.

O5. Assess joint Safety and Security (SASE) of all elements from O1, O2 and O3: Through the definition of a reference zonal vehicle architecture, this objective sets the base to ensure safety and security concerns on the creation of the cloud-connected reconfigurable zonal vehicle architecture blueprint from O1 and the resource composable SW deployment and execution layer from O2. Moreover, the combined safety-security update management and execution process (update cycle) from O3 provides a standardized approach for adopting dynamic updates, orchestration and reconfiguration, ensuring that SASE are maintained. Furthermore, reactive and preventive SASE mechanisms will be developed to prevent and mitigate safety and cybersecurity hazards that might arise due to the highly connected centralized architecture or the use of AI-based applications, reactive and preventive safety and security mechanisms will be developed. To ensure alignment with reference safety and security standards, as well as emerging regulations for SW updates, proposed concepts will be assessed by external certification authorities.

O6. Demonstration of UP2DATE4SDV solution on several CCAM relevant Use Cases: The project will develop multiple use-cases (UCs) to evaluate the results of O1-O5 and to show the value and the relevance of the developed concepts and solutions for the automotive field, especially for highly connected and automated vehicles, emphasizing on centralized architectures, reliability, cyber-security, upgradability and connectivity. Details about the use-cases and demonstrators are summarized in Sec. V.

O7. Active participation around CCAM and the SDV Initiative community and ecosystem: To acquire knowledge, reuse concepts and technologies, and foster synergies and collaborations, UP2DATE4SDV consider the reuse of building blocks from previous initiatives and parallel SDV projects, deliver building blocks to similar ecosystems (e.g. Eclipse SDV), exchange requirements beyond those explicitly assessed in the project, interact with ongoing SDV-related actions (e.g., in Chips JU), and establish a continuous dialogue to stay aligned with the main directions set by the key entities and initiatives (like FEDERATE) in the wide ecosystem.

III. Key Innovations and Results

In UP2DATE4SDV we define a number of results, many of them coming with key innovations as an outcome. The technical results innovate by increasing the Technology Readiness Level (TRL) methods, technologies, SW and HW, while the other results bring these results to the public community to share the innovation gained within the project with a broader audience.

R1. Interface- & Service-Definitions in 3 orthogonal dimensions (HAL/OAL/ORP): Clear interface definitions between our different dimensions are important to ensure a safe and secure run-time behavior following a verified specification. We will devise a resource description language to specify resources in and between the different layers, enabling the possibility to verify and monitor resource integrity during design and run-time. Close to the HW, AI accelerator platform configurations and AI accelerator performance monitoring libraries are developed with a focus on safe and secure (SASE) properties as well.

R2. Reference implementation of a reconfigurable zonal HW architecture for SD-CCAM-V: Demonstrating the application of our concepts and methods, our key results contain implementations as well. The reference implementation of a reconfigurable zonal HW architecture utilizes the new STM high performance automotive microcontroller unit (MCU) STM32MP2, a cloud-native platform for embedded applications and automotive. It is supported by an OTA service and a phase-change memory (PCM) mapper. The OTA service identifies MCUs requiring updates and uses the PCM mapper to access their memory, completing the update process on the STM Stellar MCU within the project.

R3. Reference implementation of a resource composable SW deployment and execution layer for SD-CCAM-V: For the deployment and execution layer we will enable the L4Re Operating System Framework, an open-source software (OSS), microkernel-based OS-framework and hypervisor to support standardized native safety containers, network-based updating and user-level performance monitoring. The resource-aware distributed update manager and update lifecycle management platform provide tools to add resource requirements to the specification, generation of run-time monitors and safety mechanisms to be triggered on run-time violations.

R4. Proof-of-Concept implementation of a dynamic orchestration & reconfiguration plane for SD-CCAM-V: We aim

to create a more connected, flexible, and intelligent vehicle-cloud ecosystem. We develop a dynamic orchestration and reconfiguration plane with a decentralized and distributed data fabric which provides a communication middleware unifying diverse communication protocols and storage solutions, reducing complexity and enabling seamless data flow.

R5. Safety & Security Pattern supporting certification: Supporting our SASE update approach, we will provide a reference zonal vehicle architecture pattern which allows safe and secure remote dynamic updates. The process itself is specified in a SASE update management procedure and update cycle to establish a comprehensive approach for managing updates and upgrades in compliance with the highest standards. For better integration of SASE properties into the update cycle, we will provide a SASE Mechanisms SDV Library which offers preventive and proactive safety and security mechanisms, providing diagnostics, secure communication, digital signatures, and anomaly detection, compatible with the zone-oriented architectures.

R6. Demonstrator for CCAM based collective and adaptive perception: To illustrate our research, we will develop demonstrators based on use cases that reflect the project's core challenges. These demonstrators will allow the public to understand our contributions and, in some cases, even replicate our methods for validation. See Section section V for more information.

IV. Concept and Approach

Based on four strongly connected, technical work packages, one work package for the demonstration and evaluation, and corresponding to the challenges C1-C4, UP2DATE4SDV develops solutions for the HAL, the OAL, the ORP, as well as for the safety and security of and between these layers. The following subsections summarize some of the main tasks and activities.

COTS HAL for Node Resource Monitoring and Controllability: To enable continuous SASE-aware updates, upgrades and reconfigurations across complex, heterogeneous COTS HW platforms, dedicated support at both HW and SW levels is needed. Regarding the HAL the project focuses on the HW-level features and functionalities that are necessary to support the definition of a safe and secure, modular and reusable interface that guarantees the necessary decoupling between the physical execution platform and the functionalities it offers to the upper SW layers.

While the HAL scope can vary significantly, depending on the system design and objectives, UP2DATE4SDV puts special focus on the definition and implementation of interfaces for the safe, secure, and efficient observation and control of the computation and communication resources of the computing nodes, both locally and distributed. Here, *Observability* means the ability to expose HW resources and their state in a standardized, safe, and secure way and ensures that relevant metrics (e.g., timing, CPU/memory/bus usage, fault states, etc.) are available to the OS, the orchestration, and the middleware layers. *Controllability* means the ability to allow the higher layers to control or reconfigure those HW resources and, hence, enforce corrective or adaptive actions (e.g., resource reallocation, node isolation, or firmware updates) in a safety and security compliant way. Together, observability and controllability are core functional requirements of UP2DAT4SDVs HAL as they are fundamental for assessing the system's health and performance state, and thus for validating the SASE orchestration constraints before triggering updates, node reconfiguration, the dynamic reallocation of workloads, or to enforce segregation and fault containment at run time.

Following a bottom-up, model-based approach to ensure that the HW layer exposes standardized abstractions and metrics that covers higher layers' requirements in an effective and efficient way, a second focus is on the identification and classification of an extensive, platform-agnostic set of SASE-relevant HW properties – like partitioning, redundancy, timing determinism, integrity, access control – at both intra- and inter-node levels. The identified SASE properties are then mapped to concrete monitoring and configuration mechanisms available in modern heterogeneous COTS platforms and translated into a set of metric observables for the definition of the HAL API for observability and controllability. Using the HAL API the resource requirements stemming from the SASE specifications of the OAL can be evaluated, enabling the orchestration and update mechanisms to ensure the alignment with the system's health and performance constraints without exposing the full underlying HW complexities.

Reconfigurable and resource composable SW deployment and execution layer: Facing the dynamically changing and evolving system SW layer of SDVs (O2), both the system SW components as well as the communication plane need to be able to adapt the changes in the SW and potentially the HW stack. For that, the key ingredients to success are: *Modular software design and containerization* to allow to seamlessly run partial updates of individual components of the SW stack while preserving safety and security of the overall system [6] [7]. Furthermore, this allows and necessitates a flexible and re-configurable communication plane that supports the shifting workloads of the system. *Monitoring of resource usage* such that resource constrained devices can be optimally utilized by the dynamically adapting SW and that update requests can be evaluated on real-time capacity data.

Modular system software: To allow for isolated updates of individual SW components, containerization of such components is key [8]. Switching a container due to a SW update enables keeping the rest of system as-is.

Microkernels represent the state-of-the art in the design of modular, safety-certifiable many-core systems [9]. Due to their design – with a principle-of-least-authority approach as the core design principle – modern capability-based microkernel OSs provide containerization per default. Applications only receive access to resources they have been explicitly granted. L4Re [10] is one such modern microkernel-based operating system that has previously been safety certified [11] and will be extended with a native container format within the UP2DATE4SDV project, building on previous work [12] and ongoing collaboration.

To allow communication to function seamlessly in the face of a changing SW stack the project implements a middleware (MW) for the communication plane based on named-data networking for time and resource constrained devices. This will include connecting this MW to Zenoh Flow [13] and ROS2 [14] and requires the communication plane to support the SASE mechanisms for updating the system.

Monitoring resources: To allow decisions about updating or re-allocating SW on different HW cores the project incorporates monitoring capabilities, of both the HW through performance counters as well as the SW through instrumentation, profiling and monitoring facilities. At the higher level the resource requirements will then be matched to the SoC capabilities to ensure for a valid placement of updated or shifted workloads.

System orchestration, DevOps and V&V tooling: On top of the HAL, OAL and possibly vehicle abstraction layer (VAL), the ORP coordinates the SW updates, HW upgrades and dynamic re-configurations and task re-allocations. A cloud located part of the orchestration plane integrates verfication and validation (V&V) approaches into a modern DevOps CI/CD process, providing updates targeted to vehicles and vehicle fleets. Cloud-connected, the platform located part of the ORP is then coordinating and executed the adaptation steps that were planned in the cloud on the vehicle itself. To define the architecture, interfaces and processes for the ORP, the requirements for the ORP are derived from several use cases, summarized in Sec. V.

Furthermore, UP2DATE4SDV will connect the ORP to multiple DevOps solutions in the project to integrate the update process within the DevOps cycle. First, an existing MBSE-framework will be enhanced to serve as the design and implementation framework according to the needs of the ORP, to generate reliable and certifiable code fitting to the OAL and to connect it to the V&V and design space exploration (DSE) framework in the cloud. Furthermore, this V&V framework will be developed with focus on next-generation AI-based SDV applications. It will integrate with existing DevOps and CI/CD solutions of project partners (including the aforementioned MBSE-framework) and provide OAL virtualization mechanisms on multiple abstractions levels to provide high-fidelity SDV testing environments like MIL, SIL or HIL. This way, aligned with the safety and security analysis to achieve certifiability, the DevOps process targets at testing safety and security focused non-functional properties and quality of the SW modules in addition to code-analysis.

To implement the update process, UP2DATE4SDV develops a MW extension for the vehicle to cloud continuum orchestration and deployment toolchain as well as a SASE update manager. For the cloud side orchestration, a digital twin platform with a virtual ECU (vECU) for testing and validation will be integrated. For the resource composable SW platform the project implements the synchronization with the digital twin together with a virtual integration test (VIT) for the SW composition at the update target to verify the resource compatibility during the update process. Connected to that, the update manager will implemented to support all target platforms from the use cases. The update manager will be able to download the update, to run additional on-platform safety and security checks, and will be integrated within the reconfigurable deployment and execution layer to coordinate the system adaptation.

Safety and Security Assessment: UP2DATE4SDV introduces mechanisms and processes that guarantee compliance with functional safety and cybersecurity standards while enabling continuous updates over the vehicle lifecycle. Here, ensuring that the update, upgrade, and reconfiguration processes across HW and SW layers are secure and certifiable plays a pivotal role. Consequently, the following tasks are key to achieve safe and secure updates: First, the *Update processes definition* that defines robust workflows for planning, validating, and deploying updates, including rollback and failsafe strategies to maintain system integrity under all conditions. Second, the *Reference architecture definition* that integrates safety and security mechanisms into the SDV stack, supporting containerized applications, hypervisor isolation, and MW-based configuration management. Third, *Continuous assessment and certification* through engagement with external authorities to validate compliance with commonly adopted standards (ISO 26262, ISO 24089, IEC 62443, IEC 61508).

SASE Update Process Definition: Within UP2DATE4SDV a structured update process that includes risk analysis, integrity checks, and secure key management will be defined. The update process will establish a comprehensive procedure for managing and executing updates and reconfigurations of SDV components. It will ensure systematic planning, rigorous testing, and deployment under all operational conditions, including rollback and failsafe scenarios. The task also identifies infrastructure and tools for automation, minimizing anomalies and preserving system integrity during deployment.

Reference Architecture and Mechanisms: A robust zonal vehicle architecture pattern that supports safe and secure system updates and reconfiguration will be developed. The architecture leverages containerization and hypervisor-based isolation to enforce strong separation between application domains. Middleware components manage configuration and enforce safety/security contracts. Dedicated modules implement proactive monitoring and runtime checks to maintain compliance with ASIL and SL targets.

Continuous Assessment and Certification: To ensure long-term trustworthiness, certification workflows will be integrated and external certification authorities will be engaged to validate the proposed concepts. This will ensure alignment with current and emerging safety and security regulations.

V. Use-cases and Demonstrators

In UP2DATE4SDV we define six use cases, which are implemented and shown in two demonstrators.

Research Demonstrator: Enabling early testing of state of the art technology and achievements during the project, we introduced the research demonstrator. This demonstrator is based on an affordable COTS HW platform and a OSS SW stack, using as many SW parts supplied by project partners as

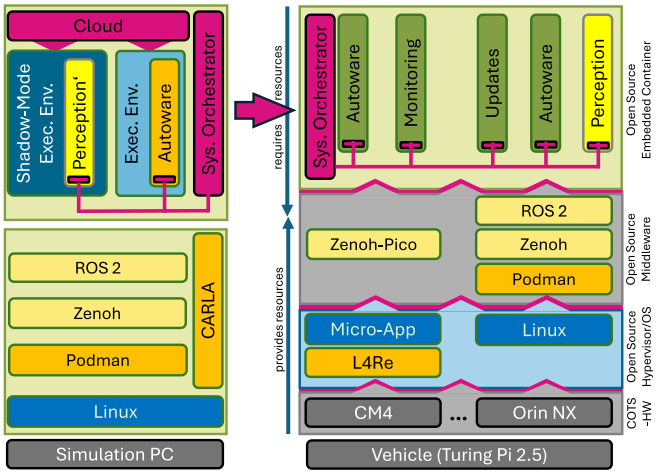

Fig. 2. The research demonstrator implements the abstract UP2DATE4SDV architecture shown before. A simulation PC is providing a HW platform for the cloud services, while the automotive part is running on Raspberry Pi CM4 and Nvidia Orin modules. The SW stacks utilizes OSS in all layers.

possible. The demonstrator is planned in such a way that it covers the more baseline and methodical use cases goals of the projects, with sub configurations for singular goals at different project partners. To allow for replication, internally as well as for publications of approaches shown on this demonstrator, we will publish a blueprint during the project.

With the research demonstrator we address four use cases: UC-1 *Updating and upgrading HW/SW in the perception chain*, UC-2 *Maintaining reliability*, UC-3 *Considering hardware resource constraints* and UC-4 *Considering Cyber-Security*. As the names of the use cases give away, the first use case has a focus on the general update and upgrade process for different HW and SW modules. For demonstration purposes we decided to use Autoware together with CARLA to run scenarios, with some parts of Autoware and the perception component running on an embedded system as target for the updates. UC-2 and UC-3 extend this setup by HW and SW monitoring, reactions to faults like module re-allocation, thus mainly focused on safety properties. UC-4 then adds a security focus to the research demonstrator as well.

For the implementation of the demonstrator we decided to use a modular HW platform, which allows to demonstrate a HW upgrade of a module while the others are still in use, as part of the goals in UC-1. The selected *Turing Pi 2.5 Cluster board* [15] allows the use of up to four modules, including the Raspberry Pi CM4 and Nvidia Jetson Orin modules we want to apply our work on. The CM4 will be mainly used to show in-place SW updates, running a type 1 hypervisor with parts of the Autoware driving function, the Autoware perception stack and middleware for execution, communication, monitoring and updating. Alongside this, we will use the Orin Nano and NX modules to show a HW upgrade, replacing the Nano with a more powerful NX module. To achieve this goal, we will also implement SW module reallocation, within modules, but also from one module to another to ensure service availability during HW upgrades. The research demonstrator, in its base variant (lacking contributions from UCs 2-4), is shown in Figure 2. It comprises a simulation PC running the CARLA environment and cloud components, alongside a Turing Pi 2.5 as the vehicle platform.

CCAM Demonstrator: Transferring the results from the research demonstrator into a realistic environment, the CCAM demonstrator extends the developments and innovations of the project by showcasing the use-cases UC-5 *Perception stack upgrade and mode switching* and UC-6 *Connected Automated Driving* via different scenarios. It will integrate automotive HPCs, a ADAS hardware-in-the-loop (HiL) system, modern verification and validation toolchains, the building blocks developed in the project and apply the SASE framework. We will first implement a virtual CCAM demonstrator for ongoing integration and testing of the project's building blocks, leveraging the open-source CARLA Simulator and Autoware perception stack. The final CCAM demonstrator will be implemented on an ADAS HiL test system using STM32MP2 and Stellar MCU devices as automotive controllers, replacing the cost-effective setup of the research demonstrator with automotive-grade components.

All scenarios are designed to highlight the benefits of the update, upgrade and reconfiguration capabilities of the proposed work. A change in ODD, sensor configuration, compute capabilities, or SW applications, need to be recognized by the in-vehicle orchestration and communicated to the cloud orchestration, to finally trigger the according OTA SW and configuration update of the vehicle in a safe and secure manner. Both the virtual and final CCAM demonstrators will showcase scenarios including: vehicle upgrades (e.g., LiDAR addition), ODD changes based on location and legal requirements, perception algorithm weight change according to weather conditions, performance-based mode switching and vehicle-to-everything (V2X) communication for threat alerts and coordinated driving (platooning) via cloud services. Besides the scenarios, the developed update process will also be demonstrated, utilizing model-based system engineering solutions for cybersecurity checks of SW updates, as well as code and task generation and resource requirement annotations for applications/services.

VI. Conclusion

UP2DATE4SDV brings together experienced industrial and scientific research players from the automotive domain to co-operatively tackle the challenges of establishing fast, seamless, and safe and secure updates, upgrades, re-configuration and task re-allocation for future SDV architectures. Together, the 12 (plus two affiliated) partners from 7 European countries are spanning the full range of knowledge and technology providers that is required to achieve the targeted objectives.

In this paper the UP2DATE4DV consortium presented for the first time its vision, challenges, objectives and key results, and drafted a first picture of its contributions within the SDV stack. At the end, the summary of the use-cases and demonstrators gives an outlook on the planned evaluation scenarios.

References

[1] H. Deubener, L. Michor, N. Patschke, and R. Heuss. (2025, Mar.) European automotive industry: What it takes to regain competitiveness. [Online]. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/european-automotive-industry-what-it-takes-to-regain-competitiveness

[2] Grand View Research. (2025) Europe automotive software market size & outlook. [Online]. Available: https://www.grandviewresearch.com/horizon/outlook/automotive-software-market/europe

[3] V. Bandur, G. Selim, V. Pantelic, and M. Lawford, "Making the Case for Centralized Automotive E/E Architectures," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1230–1245, 2021.

[4] MIT Technology Review. Self-driving cars take the wheel. [Online]. Available: https://www.technologyreview.com/2019/02/15/137381/self-driving-cars-take-the-wheel/

[5] S&P Global Mobility. More OEMs Expected to Push into Level 3 Autonomy. [Online]. Available: https://www.spglobal.com/automotive-insights/en/blogs/2025/03/more-oems-expected-to-push-level-3-autonomy

[6] N. Ayres, L. Deka, and B. Passow, "Virtualisation as a Means for Dynamic Software Update within the Automotive E/E Architecture," in *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*, 2019, pp. 154–157.

[7] P. Uven, R. Stemmer, and G. Nitsche, "A Modular Architecture Template for Resource Modeling in Software-Defined Vehicles," in *2025 20th European Dependable Computing Conference Companion Proceedings (EDCC-C)*, 2025, pp. 213–217.

[8] T. Miemietz, V. Reusch, M. Hille, M. Kurze, A. Lackorzynksi, M. Roitzsch, and H. Härtig, "A Perfect Fit? - Towards Containers on Microkernels," in *Proceedings of the 10th International Workshop on Container Technologies and Container Clouds*, ser. WoC '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 1–6. [Online]. Available: https://doi.org/10.1145/3702637.3702957

[9] K. Lampka, J. Thurlby, A. Lackorzynski, and M. Hähnel, "Safety certification with the open source microkernel-based operating system l4re," in *Computer Safety, Reliability, and Security: 41st International Conference, SAFECOMP 2022, Munich, Germany, September 6–9, 2022, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2022, p. 31–45. [Online]. Available: https://doi.org/10.1007/978-3-031-14835-4_3

[10] L4Re project members and individual contributors. The L4Re Operating System Framework. [Online]. Available: https://l4re.org/

[11] Elektrobit. (2024, Oct.) Elektrobit continues acceleration toward safe and secure software-defined mobility. [Online]. Available: https://www.elektrobit.com/newsroom/elektrobit-continues-acceleration-toward-safe-and-secure-software-defined-mobility/

[12] T. Miemietz, V. Reusch, M. Hille, L. Wrenger, J. Eisoldt, J. Klötzke, M. Kurze, A. Lackorzynski, M. Roitzsch, and H. Härtig, "MettEagle: costs and benefits of implementing containers on microkernels," in *Proceedings of the 19th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI '25. USA: USENIX Association, 2025.

[13] Eclipse Foundation. Eclipse Zenoh-Flow. [Online]. Available: https://projects.eclipse.org/projects/iot.zenoh.zenoh-flow

[14] Open Robotics. ROS - Robot Operating System. [Online]. Available: https://www.ros.org/

[15] Turing Pi 2.5 Cluster board. [Online]. Available: https://turingpi.com/