Whitepaper

Continuous Homologation for Software-defined Vehicles

digital.auto

BOSCH · ETAS · certivity · TÜVRheinland® Precisely Right. · T Systems · Ferdinand-Steinbeis-Institut · HHN Hochschule Heilbronn

Table of Contents

Introduction

This paper presents a comprehensive overview of the key concepts and challenges associated with Continuous Homologation (CoHo) of Software-Defined Vehicles (SDVs). It introduces a practical implementation framework developed by the digital.auto CoHo Special Interest Group (SIG), which includes industry leaders such as Bosch, ETAS, Certivity, TÜV Rheinland, and T-Systems, as well as research experts from Ferdinand-Steinbeis-Institute and Hochschule Heilbronn.

The framework draws on the collective expertise and real-world experiences of the CoHo SIG participants. To validate the framework, three illustrative case studies are provided. The paper concludes with a forward-looking discussion on the ongoing activities of the CoHo SIG, with particular emphasis on efforts to automate Continuous Homologation using Generative AI (GenAI).

Editor, Authors and Contributors

The following individuals have contributed to this publication:

Editor

/ Dirk Slama, Bosch and Ferdinand-Steinbeis-Institute

Authors

/ Philipp Dettling, T-Systems
/ Achim Nonnenmacher, ETAS
/ Patrick Raissle, TÜV Rheinland
/ Dirk Slama, Bosch and Ferdinand-Steinbeis-Institute
/ Nico Waegerle, Certivity
/ Marco Wagner, Hochschule Heilbronn

Contributors and Reviewers

/ Ebrahim Ameen, ETAS
/ Achim Henkel, Bosch
/ Sarunas Kondratas, NIO
/ Dominik Morar, Ferdinand-Steinbeis-Institute
/ Thang Nguyen, Bosch
/ Johannes Schild, Bosch
/ Ulrich Schulmeister, Bosch
/ Chris Seiler, Mercedes-Benz
/ Markus Sutterer, T-Systems
/ Cagatay Tuerkseven, Mercedes-Benz

Introduction

In today's rapidly evolving automotive landscape, Software-Defined Vehicles (SDVs) represent a transformative shift in how vehicles are designed, built, and maintained. Unlike traditional vehicles, SDVs rely on software to control and update key features and functions, allowing for continuous improvements and optimizations over time. As the capabilities of these vehicles evolve, so too must the processes that ensure their compliance with regulatory standards. This is where Continuous Homologation comes in—a dynamic, ongoing process that ensures SDVs meet safety and regulatory requirements throughout their lifecycle, even as software updates and new features are introduced. Given the frequent updates inherent to SDVs, maintaining compliance becomes more critical and complex, as every software change can impact performance, safety, and regulatory adherence.

Unlike traditional homologation, which typically occurs once before the vehicle is approved for the market, continuous homologation involves a dynamic, iterative process. It ensures that any software update—whether it's a minor bug fix, a security patch, or a major feature enhancement—does not compromise the vehicle's compliance with regulatory standards.

Continuous homologation is essential for ensuring that vehicles meet both OEM requirements and global regulations throughout their lifecycle. This process must be flexible enough to support both agile development approaches and traditional "V-model" value streams, which are commonly used in automotive engineering.

Dimension	Traditional Homologation	Continuous Homologation
Process Duration	Long, fixed cycles tied to model refreshes	Ongoing, allowing for continuous updates
Flexibility	Inflexible, requiring full re-certification for major changes	Highly flexible, supports incremental updates and CRs
Frequency of Updates	Infrequent, typically only during major model updates	Frequent, enabling rapid deployment of software and feature updates
Compliance Assurance	Compliance is assured at specific checkpoints	Compliance is maintained throughout the vehicle lifecycle
Cost Efficiency	High costs due to long testing and re-certification cycles	More cost-efficient by reducing re-certification needs for minor changes
Regulatory Interaction	Regulatory approval needed at each major change	Streamlined interaction, with fewer interruptions for minor updates

To effectively manage the complexity of regulatory compliance and evolving OEM standards, continuous homologation must integrate automation at every stage. Automation, including the use of advanced technologies like Generative AI (GenAI), plays a crucial role in streamlining this process. GenAI can help automate the generation of compliance documentation, test scenarios, and analysis of regulatory changes, reducing the time and effort required to maintain compliance.

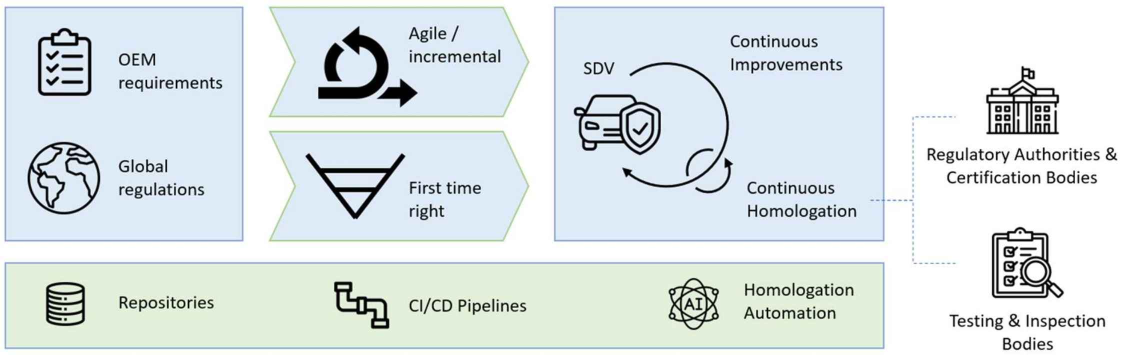

Figure 1: Continuous Homologation

By automating these tasks, continuous homologation can more easily adapt to the fast-paced, iterative cycles of agile development, ensuring that every software update or system modification remains compliant. Simultaneously, it can also support the structured, sequential processes of the V-model, where compliance is verified at each stage of development.

In the rapidly evolving automotive industry, the rise of Software-Defined Vehicles (SDVs) is reshaping traditional vehicle development and homologation processes. This paper explores the transition to SDVs and the impact on homologation. It further introduces the concept of Continuous Homologation and presents the digital.auto CoHo Framework, a structured approach to managing ongoing compliance in this dynamic landscape. Through case studies, we examine real-world applications, followed by an outlook on future developments. The paper concludes with an overview of the digital.auto CoHo Special Interest Group (SIG) driving innovation in this field.

Shifting Gears

The automotive industry is rapidly shifting gears with the introduction of new architectures and evolving value streams driven by the rise of Software-Defined Vehicles (SDVs). This transformation raises critical questions about the impact on established homologation processes. As SDVs bring continuous updates and greater reliance on software, the industry faces key challenges, including ensuring functional safety and navigating complex global regulations. How homologation adapts to these changes will define the future of vehicle compliance and certification.

Recap: The traditional Vehicle Development and Homologation Process

In the traditional automotive development process, homologation is closely tied to the V-Model, a widely used development framework that ensures both verification and validation across all stages of vehicle design. The process begins with an Initial Regulatory Analysis, where all applicable regulations for target markets are identified. This phase involves mapping out safety, environmental, and performance standards that the vehicle must comply with, varying across regions.

Once the regulatory landscape is clear, the OEM defines a detailed Homologation Strategy. This includes identifying the specific tests and assessments that must be conducted, along with timelines and resources, ensuring the vehicle meets all regulatory requirements. Next, the Test Portfolio is developed, outlining the internal and external tests that will verify compliance with safety, emissions, noise, and other standards. Testing is typically performed in-house, but often needs to be supplemented by independent 3rd Party Testing at accredited laboratories or technical services, which provide the necessary certification of compliance.

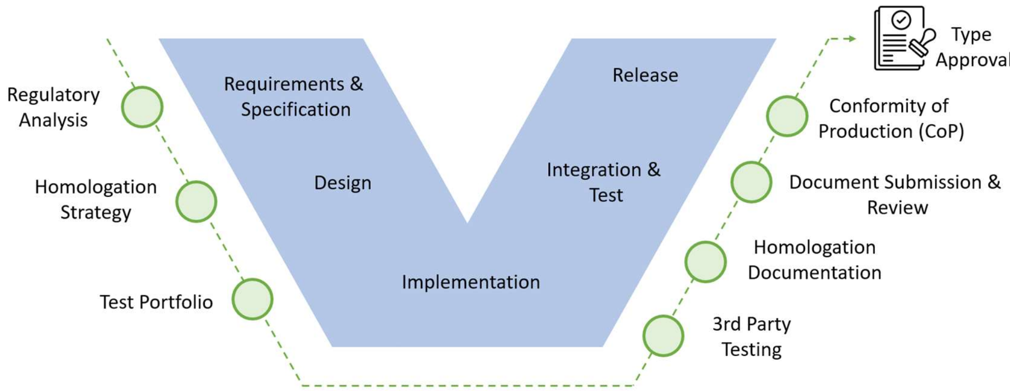

Figure 2: Traditional approach to vehicle homologation

Parallel to the testing activities, a comprehensive set of Homologation Documentation is prepared. This includes technical reports, compliance certificates, and test results, which must meet regulatory formatting and content requirements. This documentation is then submitted to relevant authorities during the Document Submission & Review phase, where regulatory bodies evaluate the data to confirm that all standards have been met.

Following document approval, the focus shifts to Conformity of Production (CoP). CoP ensures that vehicles produced in mass manufacturing continue to meet the certified specifications. Authorities may conduct audits or inspections of production processes to verify that the vehicle design remains consistent with the type-approved version.

Finally, once all testing, documentation, and CoP requirements are met, the vehicle receives Type Approval, granting it legal access to the market. This approval allows the OEM to begin production and sales, certifying that the vehicle complies with all relevant regulations for a specific market or region. The traditional process is rigorous but often static, requiring significant resources and time to meet changing regulatory landscapes, which is increasingly challenging in the age of Software-Defined Vehicles and continuous innovation.

Vehicle Development and Homologation in the Era of Software-Defined Vehicles

As the automotive industry evolves, next-gen OEMs are driving change by adopting new approaches to obtaining approvals for hardware, software, and AI updates without waiting for a full model refresh, reflecting a shift from traditional post-SOP process stability towards continuous value streams that enable ongoing innovation and optimization throughout the vehicle lifecycle.

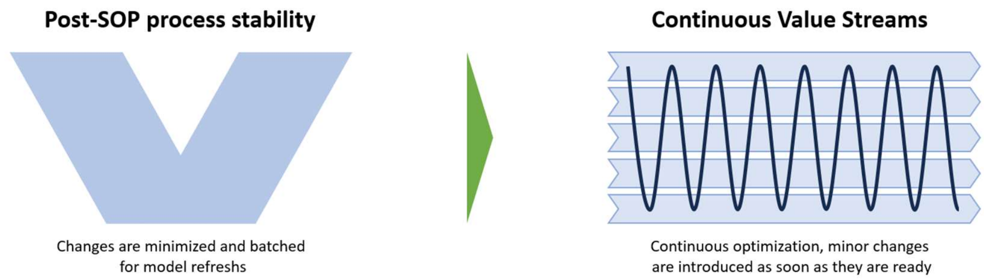

Figure 3: Shift towards continuous value streams

These companies use a combination of flexible regulatory strategies, proactive collaboration with authorities, and their software-driven vehicle architectures to streamline the approval process for all types of updates. Next-gen OEMs often seek modular type approvals for individual components, systems, and software updates. Instead of waiting for full vehicle certification, they can submit updates for specific parts—such as hardware components like sensors or electronics, or software and AI models—for regulatory approval. Minor updates that don't impact safety-critical systems or compliance, whether hardware tweaks or software optimizations, typically don't require re-certification. This allows manufacturers to notify authorities and implement updates without extensive review.

Proactive engagement with regulators is another key approach. By maintaining ongoing communication and submitting informal reviews of upcoming changes—whether hardware upgrades, software patches, or AI algorithm adjustments—next-gen OEMs help regulators assess the need for additional approvals early in the process. This reduces delays for minor updates that do not significantly alter vehicle compliance.

To further expedite the process, type approval extensions are utilized. Existing approvals are amended for minor updates, and in cases where vehicles are part of a model family, incremental changes—whether physical or digital—can be introduced without full re-certification.

A significant advantage of these OEMs is their use of over-the-air (OTA) updates. OTA capabilities allow manufacturers to remotely adjust not only software and vehicle performance but also AI models following minor hardware or software changes, maintaining regulatory compliance without requiring physical interventions.

For minor updates based on previously certified systems, next-gen OEMs often rely on certification by similarity, which allows them to bypass extensive testing by demonstrating that new hardware, software, or AI components perform similarly to those already approved.

In markets like the U.S., the self-certification system enables these automakers to implement hardware, software, or AI updates while ensuring compliance, with the responsibility for validation resting on the manufacturer. Post-production testing and OTA updates ensure any issues with AI behavior, software functionality, or hardware performance are resolved even after the vehicle is in use.

Finally, continuous integration of hardware, software, and AI updates allows next-gen OEMs to introduce incremental improvements as soon as they are ready. This provides the agility to enhance hardware components, software features, and AI systems far more frequently than traditional batch updates tied to model years. In fact, we can observe a clear shift from the long-term planning orientation of the V-Model towards continuous value streams, where updates and improvements flow steadily and in real-time, reshaping the traditional development and homologation processes.

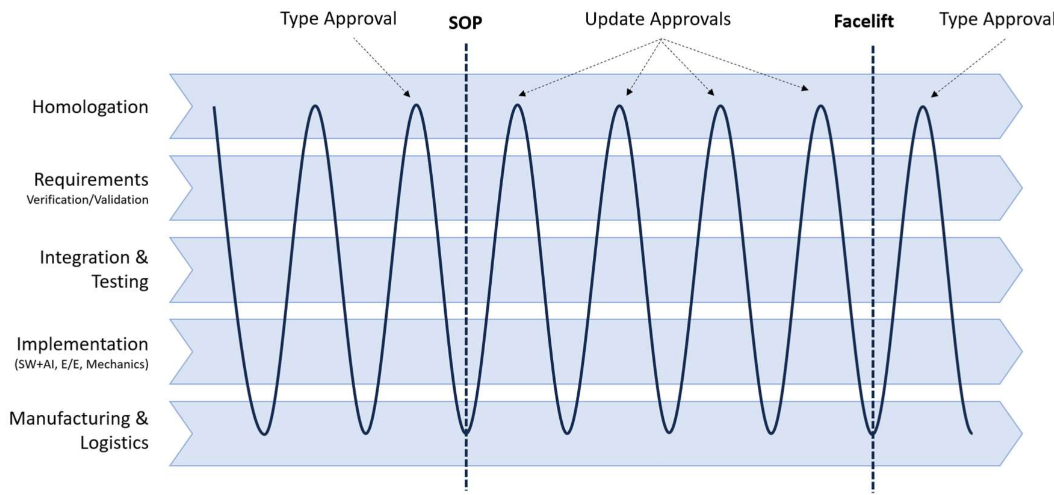

Figure 4: Continuous value streams and homologation

This holistic approach to modular approvals, proactive regulatory engagement, and continuous integration enables next-gen OEMs to implement hardware, software, and AI changes more flexibly and quickly, setting them apart from traditional automakers' slower, more rigid homologation processes.

Software Defined Vehicles: Architectures and Value Streams

A key enabler in this shift toward continuous value streams and agile homologation processes are Software-Defined Vehicles (SDVs). An SDV is a vehicle where key functions, such as ADAS, energy management, body control, and infotainment, are primarily controlled and updated through software rather than relying heavily on hardware. This approach allows for greater flexibility, enabling continuous updates and the addition of new features throughout the vehicle's life.

For ADAS, software-defined vehicles rely on complex algorithms that can be continuously refined via updates. These systems require stringent safety compliance, often utilizing embedded runtimes specifically designed for ASIL (Automotive Safety Integrity Level) functions to ensure they meet rigorous safety standards. Energy management in SDVs is similarly controlled by software, optimizing battery usage and power distribution. These systems also use embedded runtimes for safety-critical operations but can integrate with broader vehicle management software.

Body control systems in SDVs manage features like lighting and locking. These functions can be updated easily through software, and because they generally have less stringent safety requirements, they often operate in more flexible, container-based runtimes. Infotainment systems, being highly software-driven, offer features like app integration and over-the-air updates. These systems typically run in containerized environments, allowing for frequent updates and third-party app integration without impacting critical vehicle functions.

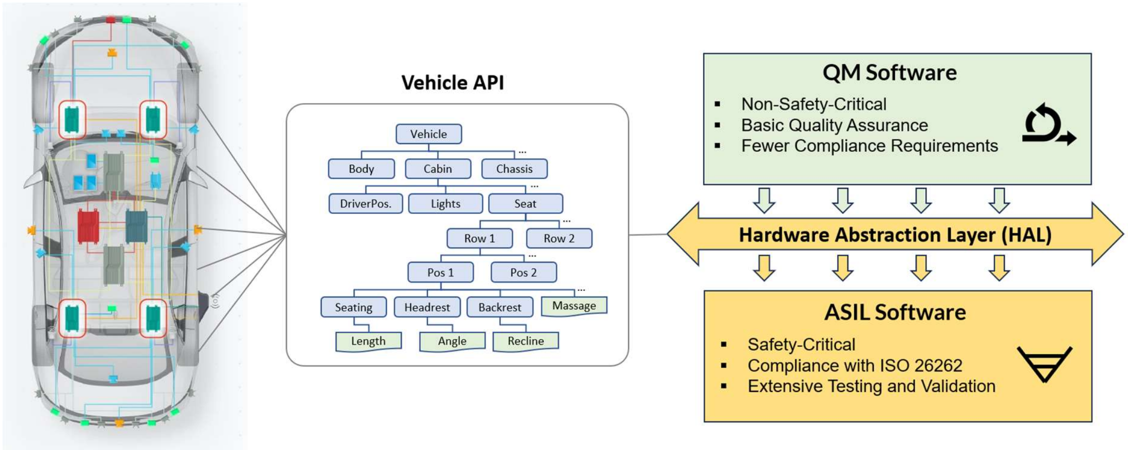

Figure 5: Software-Defined Vehicle

Vehicle APIs, such as those promoted by the COVESA initiative, are crucial in SDVs as they standardize communication between different vehicle systems and external services. Ontologies for vehicle APIs provide a structured and standardized framework that enhances interoperability and consistency across different systems, enabling seamless integration and communication within the vehicle's complex ecosystem. Specialized Signal-to-Service APIs (S2S) provide a translation from the traditionally signal-oriented word of automotive communications (e.g. via CAN bus) to the Service-oriented APIs of modern software development. This interoperability is essential for integrating new features and services from various providers seamlessly, supporting modern Electrical/Electronic (E/E) vehicle architectures. The concept of decoupling hardware from software, often through a service-oriented architecture (SOA), is key in SDVs. This approach allows different vehicle functions to be developed, deployed, and updated independently, fostering flexibility and innovation.

In SDVs, container runtimes are typically used for Quality Management (QM) functions, such as infotainment or convenience features, where they provide a flexible and updatable environment. This makes deploying and managing software updates easier. In contrast, embedded runtimes are reserved for safety-relevant ASIL functions, like those in ADAS, where reliability and strict adherence to safety standards are paramount. These environments are optimized for stability and safety, ensuring that the vehicle's critical operations remain secure and reliable.

Software-defined vehicles represent a significant shift towards software-centric design, enabling ongoing improvements and greater flexibility, with a focus on maintaining safety and performance through the appropriate use of APIs, decoupled architectures, and specialized runtime environments.

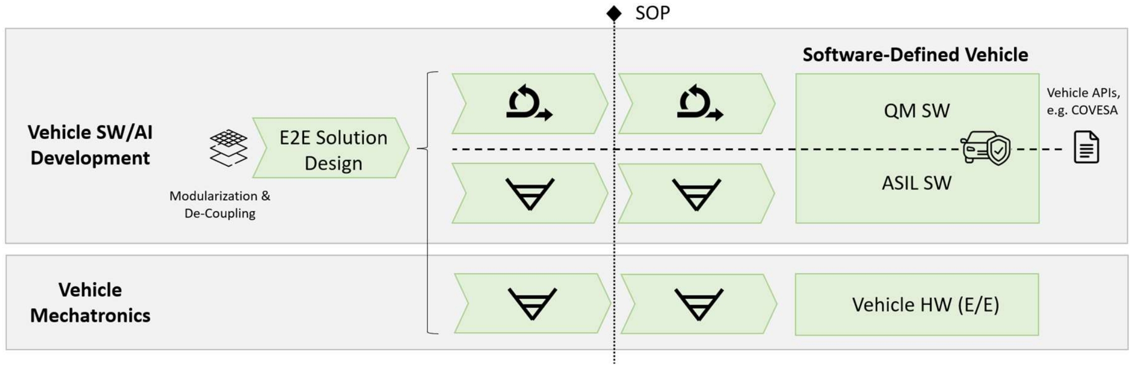

Figure 6: SDV Value Streams

Software-Defined Vehicles (SDVs) require de-coupled value streams to efficiently manage the development of both QM (Quality Management) and ASIL (Automotive Safety Integrity Level) features, each with their distinct requirements. For QM features, which are typically non-safety-critical, agile development methods, modern tools, and flexible platforms can be utilized. These allow for rapid iterations, continuous integration, and quick updates, making it easier to innovate and implement new features swiftly.

On the other hand, ASIL features—which ensure functional safety and often have stringent requirements such as hard real-time constraints—demand a more rigid development approach. For these safety-critical components, traditional methodologies like the V-Model are typically used, alongside specialized tools and platforms that guarantee compliance with functional safety standards like ISO 26262. The separation of value streams allows for the independent development of these two categories, ensuring that agile methods can thrive for non-critical features while preserving the rigor and safety assurances needed for critical systems.

Decoupling these value streams—QM features and ASIL features—is crucial for several reasons:

- Agility vs. Rigor: QM features, which are non-safety-critical, benefit from agile development methods, allowing for rapid iterations, frequent updates, and quick innovation. In contrast, ASIL features, which are safety-critical, require a highly structured, rigorous development process to meet stringent functional safety standards, such as ISO 26262. Decoupling ensures that agile methods can be applied where speed and flexibility are essential, while safety-critical features follow a more controlled, systematic approach to guarantee safety and compliance.
- Efficiency and Speed: If both value streams were tightly coupled, the entire development process would be constrained by the slowest, most rigorous requirements. By separating them, non-critical features can move forward quickly without being held back by the lengthy validation, testing, and certification cycles required for ASIL components. This approach accelerates time-to-market for non-critical updates and innovations.
- Risk Management: Decoupling reduces the risk of unintended impacts from frequent changes in QM features on safety-critical ASIL systems. It ensures that the more complex and safety-sensitive components remain stable and undergo thorough validation, while non-critical updates can evolve more dynamically without compromising the integrity of critical systems.

- Tooling and Methodology Alignment: The tools, platforms, and methods used for QM features—such as continuous integration and DevOps—are very different from those required for ASIL systems, which rely on formal verification, testing, and compliance checks. Decoupling the value streams allows each to leverage the most appropriate tools and methodologies without imposing one process on the other, ensuring both efficiency and safety.

In essence, decoupling allows OEMs to balance innovation and safety by enabling the fast-paced development of non-critical features while preserving the rigor necessary for safety-critical systems, leading to more efficient, safe, and responsive vehicle development.

Key Challenge: Functional Safety

Functional safety is a critical aspect of automotive design, ensuring that vehicles operate correctly and safely even when faults or failures occur within their systems. This is particularly important in the context of SDVs, where the reliance on software to control essential vehicle functions introduces new challenges and risks. The complexity of SDVs, with their integration of advanced electronic systems, sensors, and AI-driven software, makes it imperative to ensure that all systems function reliably under both normal and fault conditions. Failures in these systems can lead to hazardous situations, making functional safety a top priority for manufacturers and regulators.

In SDVs, the importance of functional safety is magnified by the need to manage and mitigate risks associated with software errors, sensor failures, and communication breakdowns between various vehicle components. Ensuring that safety mechanisms are in place to detect and respond to these issues is essential to prevent accidents and protect passengers and other road users. As SDVs evolve towards higher levels of automation, where the vehicle takes on more responsibility for driving tasks, the implications of functional safety become even more significant.

Automotive Safety Integrity Level (ASIL) is a risk classification system defined by the ISO 26262 standard, which assesses the potential hazards of automotive systems and their components. ASIL levels range from A to D, with ASIL D representing the highest level of risk and ASIL A the lowest. These levels help determine the necessary safety measures to mitigate risks in automotive systems.

ASIL A represents the lowest level of safety risk, where the consequences of failure are less severe, such as minor injuries or discomfort. ASIL B indicates a moderate level of risk, where failures could potentially lead to more significant injuries. ASIL C denotes a higher level of risk, where failures might result in severe injuries or potentially life-threatening situations. ASIL D is the highest risk level, where a failure could lead to catastrophic outcomes, including multiple fatalities.

In addition to the ASIL levels, the concept of ASIL QM (Quality Management) refers to situations where the risk is low enough that it does not require the rigorous safety measures specified for ASIL A-D. Instead, standard quality management processes are deemed sufficient to ensure safety.

Global Regulations

Global regulations related to functional safety in cars are not only numerous but also vary significantly in complexity. **ISO 26262**, the primary international standard, is highly detailed, covering everything from system development and hardware design to software validation and testing. It requires thorough risk assessments, fault tree analysis, and compliance with stringent safety integrity levels (ASIL). This standard alone can be highly complex, demanding significant resources and expertise from OEMs.

UNECE Regulation No. 156 (R156) is the specific regulation under the United Nations Economic Commission for Europe (UNECE) framework that governs **Software Update Management Systems (SUMS)** and the use of **Rx Software Identification Numbers (RxSWIN)** in the automotive sector. R156 establishes requirements for managing software updates throughout a vehicle's lifecycle, ensuring that updates are safe, secure, and compliant with regulatory standards. It mandates that manufacturers implement a SUMS to control, document, and trace software changes in vehicles, enhancing safety and cybersecurity. Additionally, R156 requires the use of RxSWIN as a unique identifier for each software version in Electronic Control Units (ECUs), enabling precise tracking and verification of software during type approval, conformity of production, and in-service operations. Together, these measures ensure transparency, traceability, and regulatory compliance in the automotive software ecosystem.

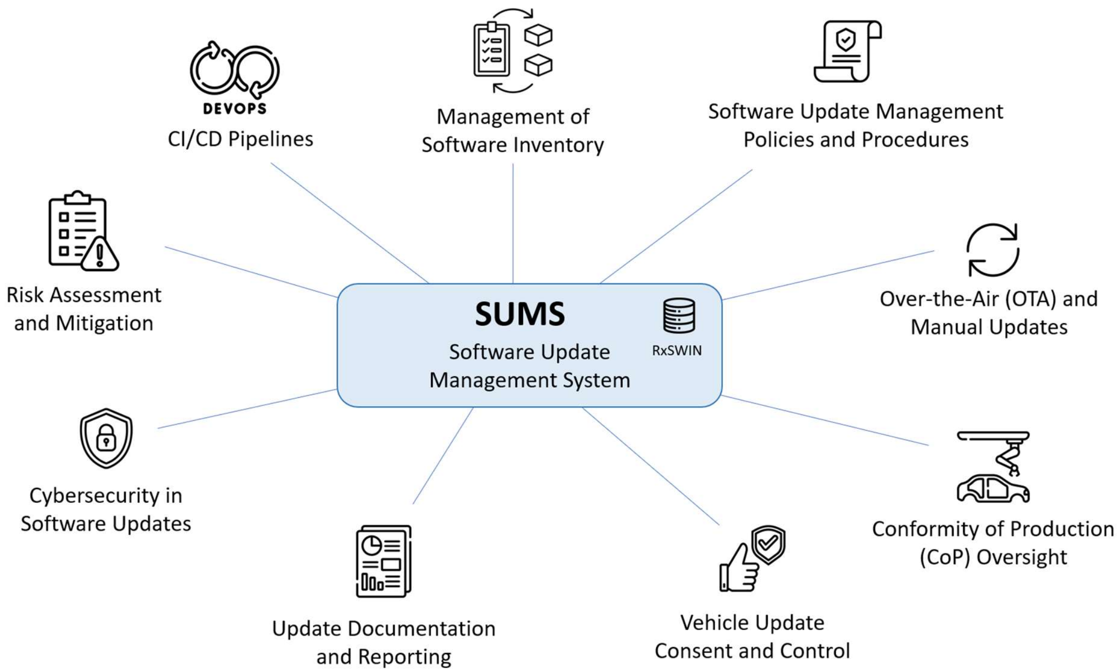

Figure 7: Software Update Management System (UNECE R156)

For ADAS, regulations like **UNECE R79** (steering equipment) and **R157** (automated lane keeping system) involve rigorous testing and validation protocols. They require precise performance metrics and real-world testing scenarios to ensure systems like lane-keeping and automated driving are safe under various conditions. These regulations also often necessitate compliance with several sub-standards or guidelines, each adding layers of complexity

Energy management regulations, such as **UNECE R100** and **SAE** standards, involve detailed requirements for battery safety, thermal management, and crashworthiness. These regulations require sophisticated testing procedures and often involve advanced simulation and modeling to ensure compliance.

Body control regulations, while sometimes less complex than ADAS or energy management, still require careful consideration, especially as these systems integrate more with other vehicle functions. For example, **UNECE R48**, which covers lighting systems, includes specific technical requirements for the design, installation, and operation of lighting and light-signaling devices, which must be carefully met to avoid non-compliance.

Overall, for a new vehicle type, an OEM must navigate through 100-200 regulations, each with its own set of detailed requirements and complexities. For a specific vehicle feature like ADAS, the 5-10 regulations involved are often highly complex, requiring extensive validation and certification processes. Similarly, energy management and body control features involve 2-7 specific regulations, each demanding in terms of technical detail and compliance procedures. The combination of the number and complexity of these regulations makes the regulatory landscape in the automotive industry particularly challenging.

Implementing Continuous Homologation

We are now exploring Implementing Continuous Homologation, starting with a deeper understanding of how Change Requests impact compliance in dynamic vehicle environments, leading into a broader analysis of the processes required to manage regulatory approvals effectively.

Change Requests and Software Updates

In the context of Continuous Homologation, Change Requests (CRs) play a pivotal role in ensuring that vehicles remain compliant and up-to-date across their lifecycle. A CR is not simply a request for a minor modification; it represents a detailed and structured mechanism for managing changes across all aspects of a vehicle, both digital and physical. This includes addressing all vehicle requirements—functional, non-functional, and regulatory—along with specific components that may be impacted by the change.

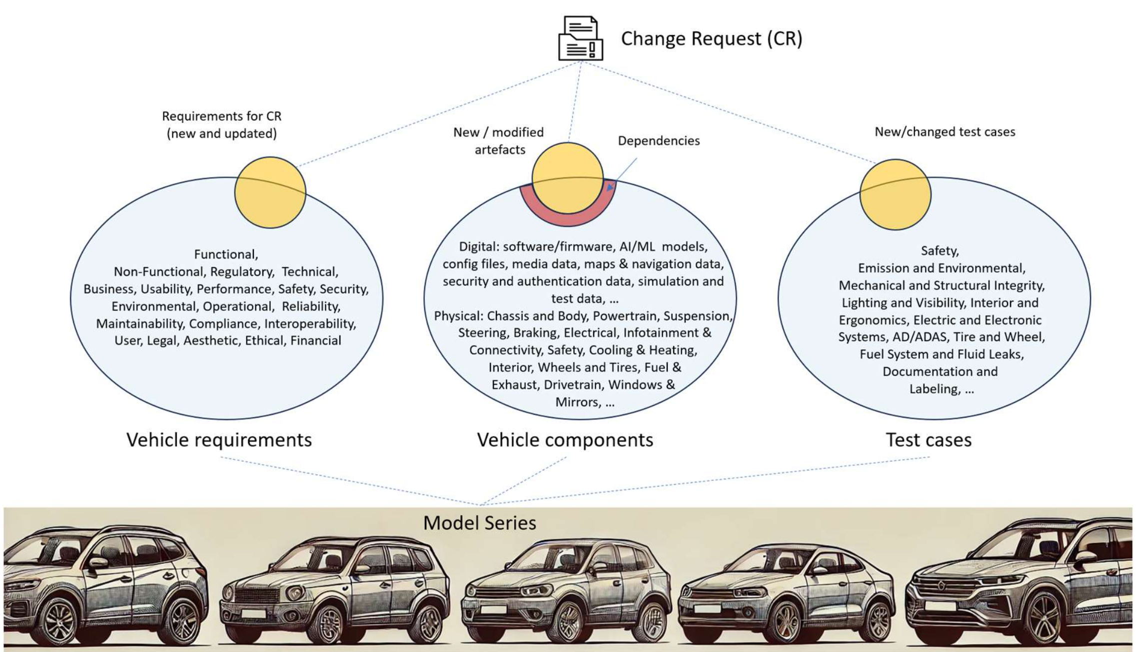

Figure 8: Change Requests and Software Updates

CRs typically apply to an entire model series, meaning the proposed changes must be evaluated not just for a single vehicle but for all vehicles within that series. This evaluation includes the assessment of all test cases that are linked to the affected components and systems, ensuring that any updates maintain the integrity of both safety-critical and non-safety-critical functions. Whether the change involves hardware, software, or AI systems, the CR process must ensure that all regulatory and homologation requirements are met for every relevant market, making it a comprehensive and critical process in the modern automotive landscape.

Given the increasing complexity of Software-Defined Vehicles (SDVs), managing CRs effectively is essential to ensure that updates—whether for features, bug fixes, or regulatory compliance—can be smoothly integrated into the vehicle ecosystem. Each CR must be thoroughly analyzed to assess its impact on interconnected systems, ensuring that no component or requirement is overlooked. This holistic approach allows for continuous updates, aligning with modern development methodologies while maintaining compliance with global regulations.

High-level process definition

Managing a Change Request (CR) effectively in the context of Continuous Homologation involves a structured and iterative process to ensure that all aspects of the change are thoroughly evaluated and compliant with the necessary regulations.

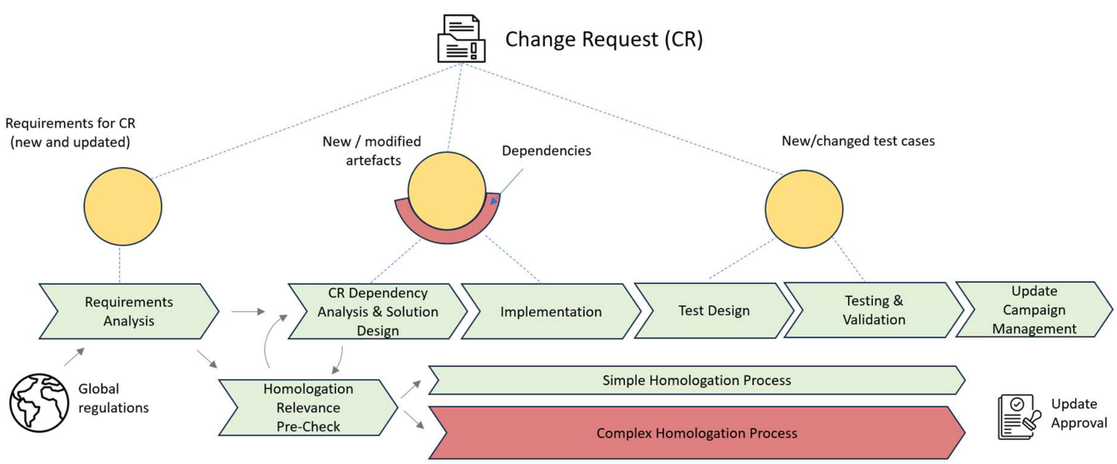

Figure 9: CR management and Continuous Homologation

The process begins with Requirements Analysis, where all the impacted vehicle requirements—whether functional, non-functional, or regulatory—are identified. A crucial part of this step is recognizing which regulations apply to the proposed change, as different regions may have varying standards.

Following this analysis, the process moves into a cycle of CR Dependency Analysis, Solution Design, and Homologation Relevance Pre-Check. This iterative phase ensures that each proposed solution accounts for all dependencies within the vehicle's architecture, including both digital and physical components. The homologation pre-check ensures that any regulatory implications of the change are considered early in the process. This cycle continues until a final design decision is reached and the appropriate homologation process is triggered.

Depending on the complexity of the proposed change, the homologation process can either be relatively simple—requiring minimal regulatory interaction—or more complex, involving extensive testing and documentation. Once the CR is finalized, the CR Implementation phase begins, followed by the definition of a suitable test suite that will validate the change.

After testing and validation, the results are shared with the relevant regulatory authorities as part of the homologation process. Once update approval is acquired, an update campaign can be initiated, deploying the approved changes across the relevant vehicle series.

Throughout this entire process, the Software Update Management System (SUMS) plays a crucial role in managing and documenting all system states, ensuring that each update is properly tracked and compliant with regulatory standards. SUMS ties together the various phases, providing a transparent and controlled environment for continuous updates in modern vehicles.

Identifying and managing relevant regulations

Identifying and managing relevant regulations is one of the most challenging aspects of the homologation process, particularly for Software-Defined Vehicles (SDVs). The complexity arises from the fact that regulations vary significantly across different countries and regions, often leading to a tangled web of requirements. Regulatory standards are issued by various agencies, each maintaining their own versions of documents, which frequently reference other standards and may change over time. This creates a highly heterogeneous landscape of regulatory documents, making it difficult to determine which rules apply to a given Change Request (CR).

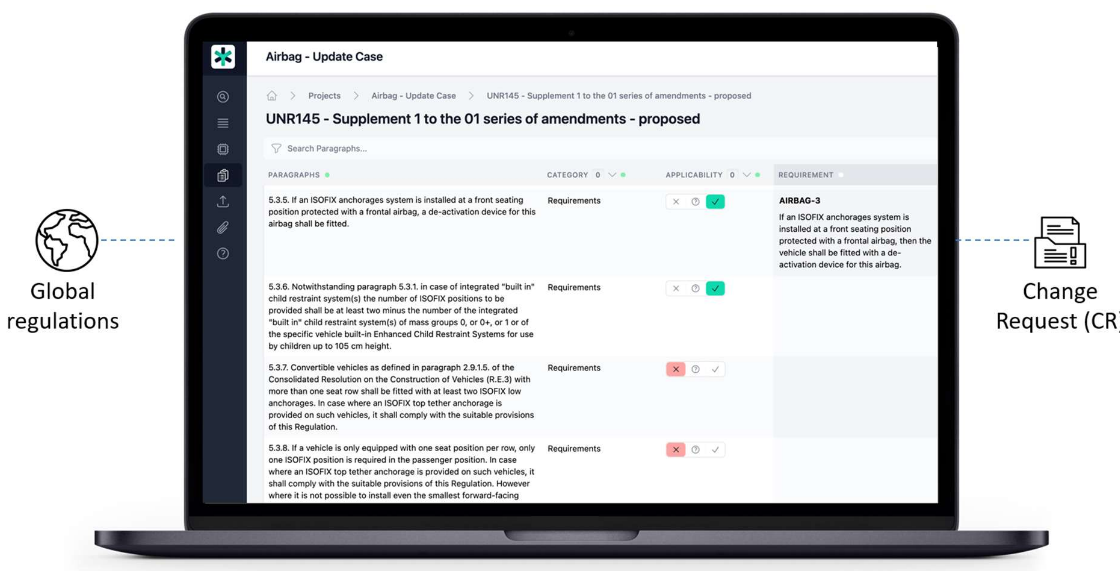

Figure 10: Identifying and managing relevant regulations

To navigate this complexity, the use of emerging Regulatory Database (RegDB) tools, like Certivity (one of the contributors to this SIG), can be highly beneficial. These tools provide a centralized, well-structured repository of global regulations, offering a clear and organized view of the regulatory landscape. With all relevant regulations housed in one place, these tools streamline the process of identifying applicable requirements across multiple jurisdictions.

Moreover, RegDB tools can help match CRs with relevant regulations, allowing manufacturers to create a CR-specific set of regulatory requirements. This tailored set of regulations can serve as input during the Homologation Pre-Check process, ensuring that each CR is evaluated against the most up-to-date and relevant standards. By simplifying access to and understanding of complex regulatory documents, these tools can significantly reduce the time and effort required to ensure compliance, while also minimizing the risk of overlooking critical requirements.

CR solution design and dependency analysis

Once the Homologation Pre-Check has provided an initial assessment of regulatory requirements and constraints, the CR solution design phase becomes critical. The design must not only ensure full regulatory compliance but also account for cost, feasibility, and other practical factors such as usability and user experience (UX). The outcome of the homologation pre-check significantly influences how the design is approached, especially when multiple design options are available. Balancing these factors is essential in selecting a solution that aligns with both technical and business objectives.

At the heart of this process is ISO 26262 dependency analysis, which plays a key role in ensuring that the proposed solution adheres to stringent functional safety standards. ISO 26262 is the global standard for automotive functional safety, and it defines requirements for managing risk and ensuring that safety-critical systems perform reliably in the face of potential hazards. In the context of CR solution design, this dependency analysis focuses on identifying and evaluating all the interactions between safety-critical and non-safety-critical components that could be affected by the change.

The dependency analysis involves understanding how the proposed solution interacts with other systems and components across the vehicle architecture. This includes mapping out any potential failure modes or cascading effects that might arise from the change. By carefully analyzing dependencies, the solution can be designed to mitigate risks and avoid introducing safety concerns, particularly in systems governed by ASIL (Automotive Safety Integrity Level) classifications, which range from ASIL A (low criticality) to ASIL D (highest criticality).

In some cases, ISO 26262 decomposition can be employed to simplify the certification process for certain elements of the system. Decomposition allows the partitioning of a complex system into smaller, independent units, each of which can be developed and verified separately. This reduces the overall safety integrity level required for certain components while ensuring that the overall system maintains the necessary safety standards. For example, by isolating an ASIL D component from a less critical QM component through hardware and software separation, the development process can be streamlined without compromising safety.

For instance, if a CR involves updating software that interacts with both QM (Quality Management) and ASIL components, the dependency analysis would focus on ensuring that the update does not compromise the integrity of the safety-critical systems. This may include evaluating the impact on hard real-time systems that operate under strict timing constraints, ensuring that any change maintains the required performance characteristics.

Ultimately, the solution design must account for both safety and practical considerations. Once compliance with regulatory and functional safety standards has been ensured, cost-efficiency, technical feasibility, and UX can guide the final design decision. By conducting thorough ISO 26262 dependency analysis and integrating the findings from the homologation pre-check, along with leveraging ISO 26262 decomposition when applicable, the solution can be implemented in a way that balances compliance, safety, and operational efficiency.

The digital.auto CoHo Framework

The following introduces the digital.auto Continuous Homologation Framework and offers a detailed guide on establishing the necessary infrastructure and processes.

Framework Overview

The digital.auto Continuous Homologation Framework integrates end-to-end solution design principles for Electrical/Electronic (E/E) architectures and Software-Defined Vehicles (SDVs) with specialized work streams, both agile and traditional, while incorporating rigorous requirements management, testing, and vehicle homologation processes. This framework ensures that every aspect of vehicle development, from initial design to post-production updates, is meticulously managed and compliant with global regulations.

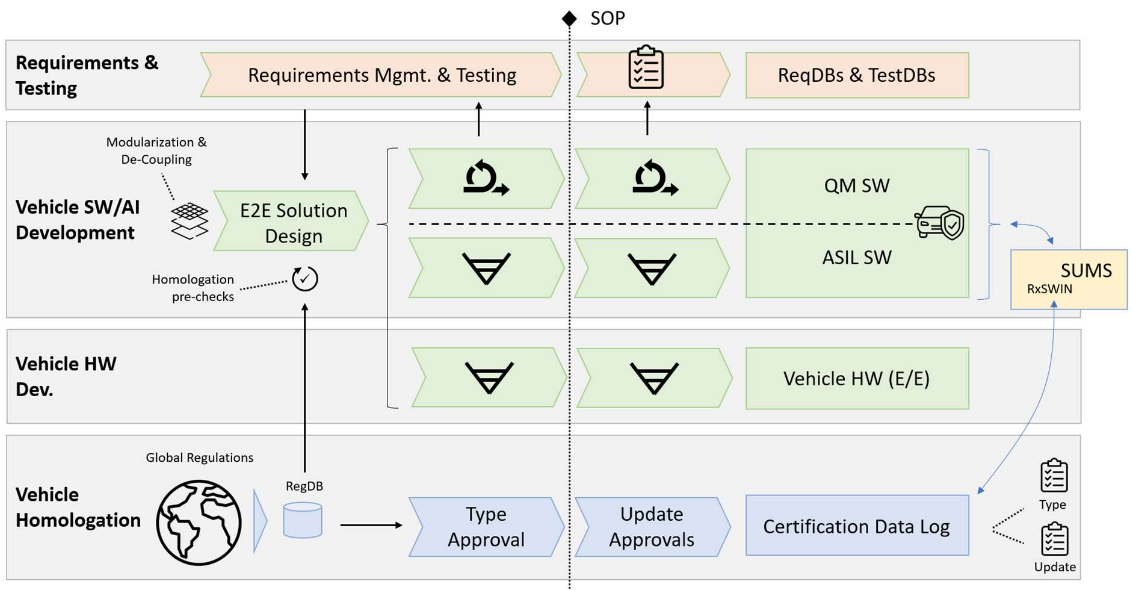

Figure 11: digital.auto CoHo Framework

At the heart of this framework is a Requirements Management System paired with a corresponding Test Database. These systems track and align requirements with their respective tests, ensuring that every component and function of the vehicle meets the necessary standards throughout its lifecycle. This traceability is crucial as it allows for precise verification that all design and safety requirements are consistently met.

Vehicle homologation within this framework is driven by a Regulation Database (RegDB), which contains structured and detailed information about global automotive regulations. This database is essential in feeding accurate and up-to-date regulatory information into the design and testing process, ensuring that the vehicle complies with the diverse regulations across different markets. The RegDB can be used to perform the required homologation pre-checks as part of the end-to-end system design, and to support along the detailed design, development and approval phases.

The homologation process starts with Type Approval, where the vehicle initially receives approval for a specific type, confirming its compliance with all necessary regulations. However, unlike traditional processes, the digital.auto framework acknowledges that development continues beyond the Start-of-Production (SOP). Post-SOP, Update Approvals are required for any software or hardware modifications, ensuring that the vehicle remains compliant as it evolves.

Supporting this continuous compliance effort is the Certification Data Log, which meticulously records all necessary information related to the vehicle's compliance and certification status. This log is a crucial element of the Software Update Management System (SUMS), providing a comprehensive and accessible record of the vehicle's certification history.

Finally, RxSWIN IDs (Regulated Software Identification Numbers) create a critical link between software artifact updates and the Certification Data Log. These IDs ensure that any software changes are accurately tracked and mapped to the vehicle's certification status, maintaining a clear and auditable trail of compliance for each update.

In summary, the digital.auto Continuous Homologation Framework seamlessly integrates end-to-end solution design with regulatory compliance, supporting ongoing development and ensuring that vehicles remain safe, compliant, and up-to-date throughout their entire lifecycle.

Simple vs Complex Homologation Process

The digital.auto framework for Continuous Homologation is designed to streamline and optimize the homologation process by categorizing it into two distinct setups: Simple Homologation and Complex Homologation. This dual approach enables organizations to minimize homologation efforts and resources by tailoring the process based on the specific relevance and impact of introducing new features or changing existing ones. The choice between Simple and Complex Homologation setups is determined by a homologation relevance check at the feature level, which assesses the implications of the changes on the vehicle's overall architecture and safety requirements.

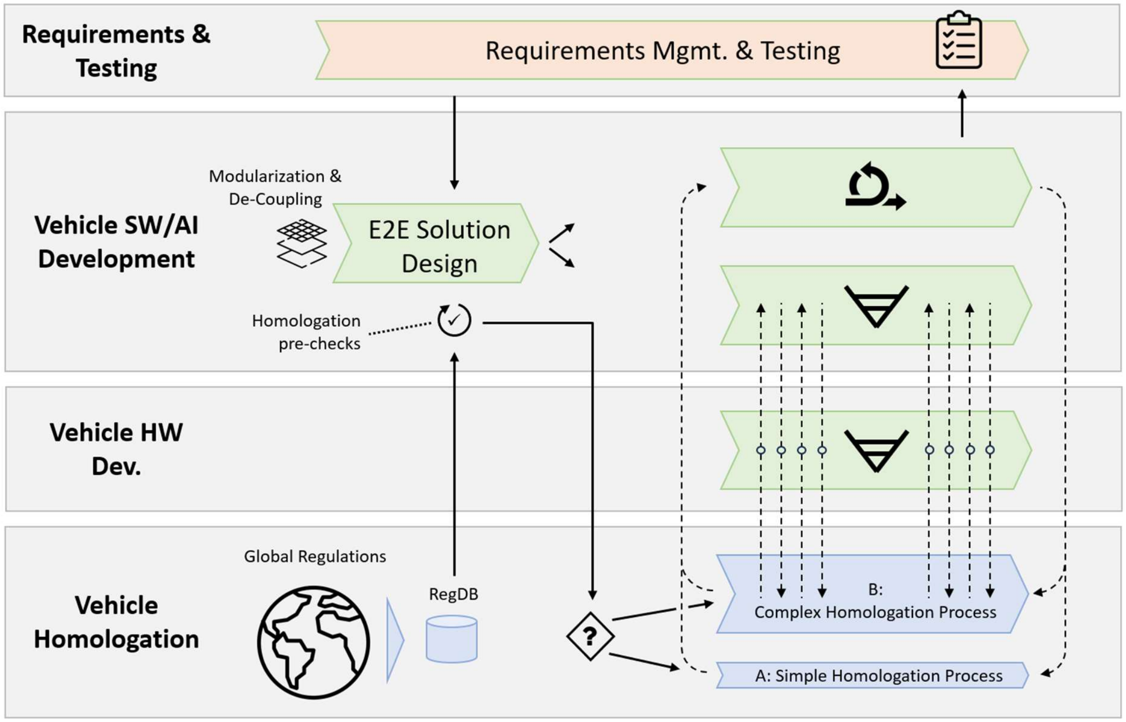

Figure 12: Homologation Process Selection

The Simple Homologation setup is intended for new features or changes that exclusively affect Quality Management (QM) software components. Since these components typically do not directly impact critical safety functions, the homologation process can be streamlined. Simple Homologation usually involves only a pre-check and final confirmation during the release cycle, ensuring compliance without extensive cross-disciplinary collaboration. This approach is both cost-effective and time-efficient, making it ideal for frequent updates and iterative development cycles.

The Complex Homologation setup, on the other hand, is mandatory for new features or changes that influence Automotive Safety Integrity Level (ASIL) software or hardware components. Given the critical nature of ASIL components in ensuring vehicle safety, this setup requires a far more comprehensive and coordinated approach. Software designers, hardware experts, and homologation professionals must work closely together throughout the entire process—from detailed component design to final approval. This ensures that all safety-critical elements are rigorously analyzed, validated, and documented to meet stringent regulatory requirements.

Dimension	Simple Homologation	Complex Homologation
System Dependencies	Low system dependencies, typically limited to non-safety-critical components	High system dependencies, often involving safety-critical ASIL components
Regulatory Requirements	Basic compliance with QM standards, minimal regulatory engagement	Extensive regulatory requirements for functional safety, ASIL compliance

Homologation Process	Streamlined, involves pre-checks and final confirmation	Comprehensive and coordinated, end-to-end process
Stakeholder Involvement	Minimal cross-disciplinary involvement, usually limited to QM teams	High level of involvement, including software, hardware, and homologation experts
Testing Complexity	Low complexity, often limited to signal or basic functional tests	High complexity, requiring rigorous testing and validation
Compliance Documentation	Simple and minimal documentation requirements	Detailed and extensive documentation to meet global regulatory standards
Cost	Lower cost due to reduced process and resource involvement	Higher cost due to comprehensive process and cross-functional involvement
Time-to-Market	Fast, designed for frequent and iterative updates	Longer time frame due to the need for thorough validation and testing

A key strategic objective for a Software-Defined Vehicle (SDV) architecture is to progressively "shift north" more functionality into the QM domain, enabling the use of the more efficient Simple Homologation setup for a broader range of updates and changes. This reduces reliance on the more resource-intensive Complex Homologation process, supporting a more agile and cost-effective development workflow.

To achieve this, the framework emphasizes the introduction of service architectures, hardware abstraction, and loose coupling between software and hardware components. By decoupling these elements, safety-critical functionalities can be isolated, allowing the majority of updates and changes to be classified as QM rather than ASIL. This architectural approach not only reduces homologation costs and time but also accelerates innovation cycles, enabling continuous delivery in the automotive sector.

Case Studies

The following section introduces three case studies: "Passenger Welcome Sequence," "Acoustic Vehicle Alerting System (AVAS)," and "Airbags & Baby Seat Detection." These case studies serve to validate the core principles of the digital.auto Continuous Homologation Framework, illustrating its applicability in diverse real-world scenarios.

As we will explore, each case study presents distinct challenges in terms of system dependencies, the difficulty in measuring the impact of changes, and the overall complexity of the homologation process required. These differences will be further analyzed following the introduction of each case study, highlighting the adaptability and robustness of the framework in varying contexts.

Simple Homologation: Passenger Welcome Sequence

The "Passenger Welcome Sequence" is an example of how Continuous Homologation manages innovative features that enhance user experience while ensuring compliance with regulatory standards. In this scenario, the vehicle detects the proximity of an approaching passenger using advanced sensors. Upon detection, the vehicle initiates a series of actions: the door automatically opens to allow seamless entry, a welcome light sequence illuminates the pathway and key areas like door handles and footwells, and the seat adjusts according to the passenger's pre-set preferences. See Figure 13 for an example implementation in the digital.auto playground.

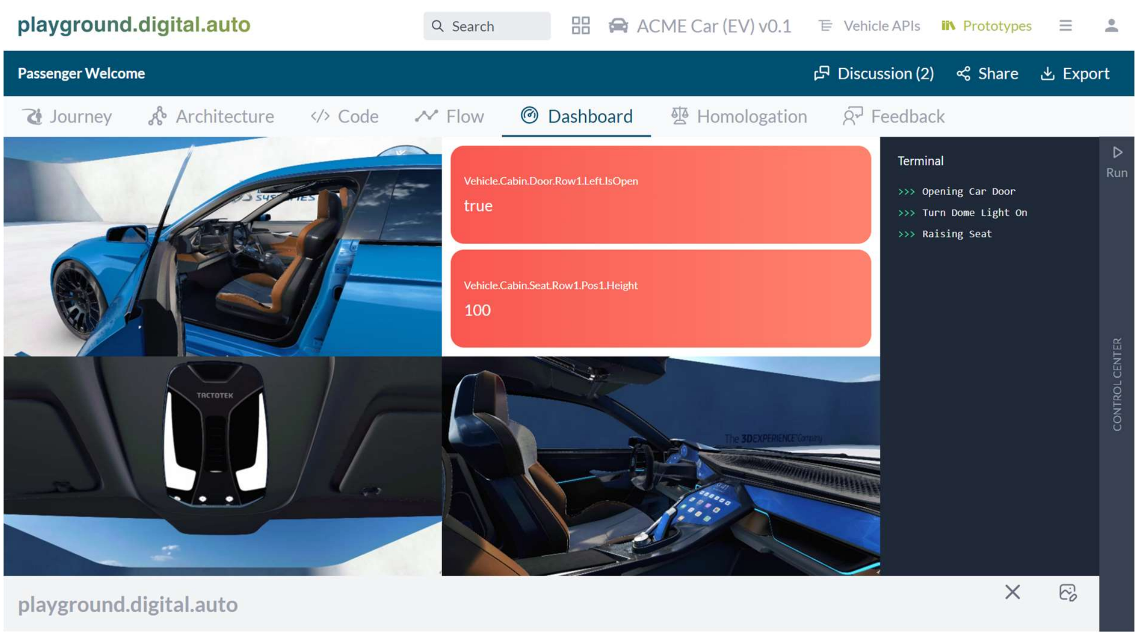

Figure 13: Passenger Welcome Sequence in the digital.auto playground

This sequence involves several interconnected systems, including proximity sensors, automatic doors, lighting, and seat controls. Any updates or modifications to these components, such as improvements to sensor algorithms or changes in the lighting configuration, must be carefully managed to ensure ongoing compliance with safety and regulatory standards.

One key vehicle API domain utilized in this example is vehicle doors. For example, COVESA VSS is defining a set of APIs to monitor and control different functions of a vehicle door, for doors in different positions (row 1 vs 2, left vs right). See Figure 14 for examples.

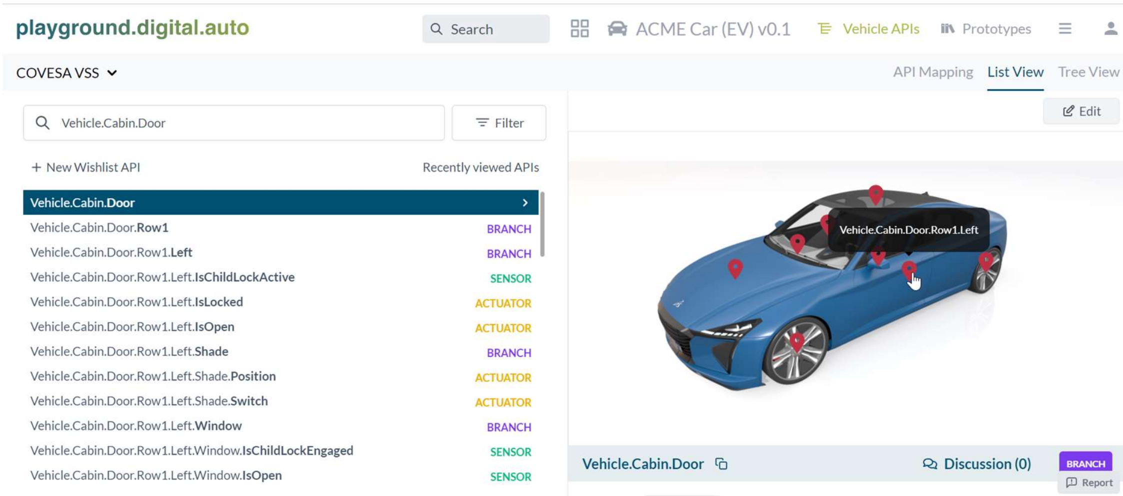

Figure 14: Vehicle doors as COVESA Signal-to-Service APIs

Since COVESA VSS is based on a semantically rich ontology for structuring the vehicle API tree, it is possible to utilize COVESA VSS API names as a search query against a regulatory rules DB. Figure 15 shows an example in the digital.auto playground, utilizing Certivity as the RegDB.

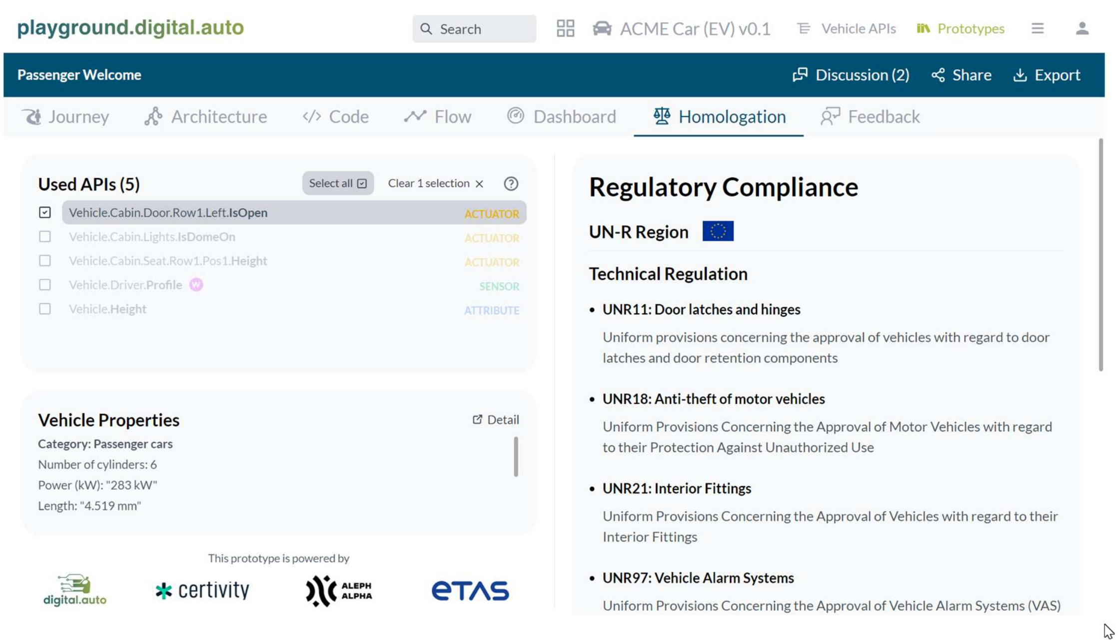

Figure 15: Matching COVESA VSS ontology elements with regulatory requirements

In this example, the Passenger Welcome Sequence is implemented as a QM function, operating within a QM container runtime on-board the vehicle. This sequence utilizes the door API to automatically open the door as the passenger approaches. It is crucial to note that the door API itself is implemented within an ASIL runtime environment, ensuring that safety-critical aspects are rigorously managed.

For instance, the open door API must guarantee that the door only opens when the vehicle is stationary, preventing any accidental door openings while the vehicle is in motion. Additionally, the system integrates cameras and AI functions to assess the environment before opening the door. These AI-driven checks ensure that there are no moving objects with potentially colliding trajectories and that no stationary obstacles are obstructing the door's path.

In addition, the use of the open door API has to be protected so that only authorized users can open the door. This is, for example, mandated by the following regulations:

- UNR116 outlines anti-theft and immobilizer requirements for motor vehicles
- UNR161, UNR162, and UNR163 provide specific standards for various components and systems related to vehicle security, including keyless entry, alarm systems, and electronic controls to enhance vehicle theft prevention and occupant protection.

In our example implementation, we are assuming that the underlying middleware supporting the door API (e.g. Eclipse KUKSA) is performing a check to ensure that only authorized users can open the door.

Figure 16 shows how a version 1.0 could look like, including an implementation of the passenger welcome sequence which is utilizing a safe implementation of open door. This whole system setup will undergo a homologation check, as per our previous discussion. The approvals of both the welcome sequence as well as the open door API implementation will be recorded with their respective RxSWIN IDs in the Certification Data Log.

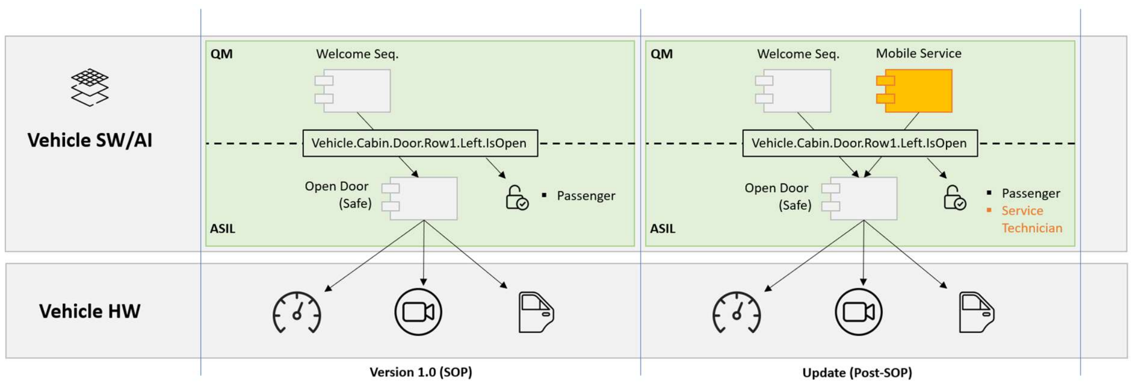

Figure 16: Open door API and related use cases

Now, in the next iteration of this use case – after SOP – the OEM decides to add a Mobile Service application. This application allows customers to make service appointments, e.g. for regular vehicle service checks, changing tires, or for fixing minor problems with a vehicle. The app will schedule an appointment with a service technician, who will perform the service on the parked car, without the car owners assistance or even presence. In order to do so, the service technician will utilize an app which will enable him to open the car, without support from the owner. In Figure 16, the required changes for this update are highlighted in orange: a new Mobile Service app is added, and a new role "Service Technician" has been added to the authorization service. No changes have been made to any code residing in the ASIL runtime, including the "open door" implementation.

Consequently, the approval of the new Mobile Service application should be possible without checks to any existing components in this environment. The safe implementation of the open door API is an isolated component, which is not changed for this update. Thus, it does not have to be approved again.

This should have been clearly identified during the homologation pre-check. Consequently, the team should have opted for the Simple Homologation Setup. During the release phase, this will only require a confirmation that the initial findings which have been made during the homologation pre-check have been confirmed during the implementation phase.

Simple Homologation: AVAS

The Acoustic Vehicle Alerting System (AVAS) is a safety feature required for electric and hybrid vehicles, which are often much quieter than traditional internal combustion engine vehicles. The purpose of AVAS is to generate sound at low speeds to alert pedestrians and cyclists to the presence of a vehicle, thereby reducing the risk of accidents. Regulations mandate that the system emit sound when the vehicle is traveling below a certain speed, usually around 20 km/h, and as the vehicle reverses.

AVAS is regulated by various global standards. For example:

- In the European Union, UN Regulation No. 138 sets the standard for AVAS, requiring vehicles to emit sound between 56 and 75 decibels (dB) when traveling at speeds below 20 km/h. The sound should be continuous and increase with vehicle speed.
- In the United States, the Pedestrian Safety Enhancement Act mandates that all electric and hybrid vehicles must emit an audible sound when traveling at speeds below 30 km/h.
- In China, the requirement is slightly more stringent, where the sound level is mandated to be at least 60 dB, slightly higher than in other regions.

The AVAS use case provides a good example of another Simple Homologation process. AVAS is primarily a Quality Management (QM) feature since its purpose is not safety-critical in the sense of directly impacting driving performance or functional safety, but it still has regulatory relevance for compliance. The system operates independently from ASIL components such as braking or steering systems, which means that the homologation process can focus on verifying the system's functionality without requiring a full-scope homologation process.

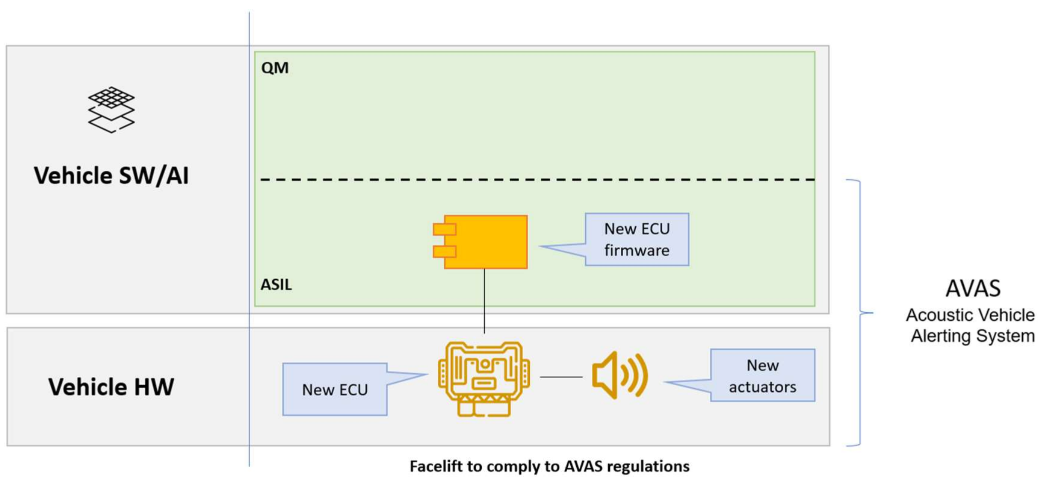

Figure 17: AVAS Case Study

The homologation process for AVAS is simplified because it involves straightforward compliance checks such as sound level testing, ensuring that the emitted sound fits within the regulatory parameters for volume and frequency range. Since AVAS is a stand-alone system with limited interaction with other critical vehicle components, the process does not require extensive cross-discipline coordination or complex testing.

AVAS validation can be achieved using relatively simple signal analysis techniques. A sound test is conducted to ensure that the system produces sound at the correct decibel levels under various conditions. Engineers can analyze the frequency, amplitude, and duration of the sound emitted by the system to confirm it meets regulatory requirements. For example, a sound meter can be used to measure the decibels issued by the AVAS when the vehicle is in motion or reversing, ensuring compliance with local regulations.

The specific requirements for AVAS differ slightly between regions, which can affect the homologation process. For instance:

- China requires AVAS to emit sound at a slightly higher minimum decibel level (60 dB) compared to Europe, where the minimum level is 56 dB. This difference means that a vehicle configured for European markets may need adjustments or validation testing when being homologated for the Chinese market.
- Additionally, the frequency and sound profile of AVAS may be subject to different regulatory expectations depending on local pedestrian and environmental noise standards.

While these variations may necessitate additional validation tests when entering new markets, the overall Simple Homologation process remains applicable due to the limited system dependencies and the clear, measurable compliance criteria for AVAS.

By managing the homologation process through simple signal analysis and focusing on region-specific requirements, AVAS demonstrates how the digital.auto Continuous Homologation Framework can be applied to efficiently handle minor but important regulatory features.

Complex Homologation: Airbags & Baby Seat Detection

The third case study we introduce exemplifies a Complex Homologation Setup, highlighting the complexities involved in homologating safety-critical systems in modern vehicles. The use case centers on the automatic detection of a baby seat placed in the front passenger seat next to the driver. In the initial version of the vehicle's software, Artificial Intelligence (AI) is used solely to provide a warning when a baby seat is detected, recommending that the driver manually disable the front passenger airbag. However, in the next iteration, this function is set to evolve: upon detection of a baby seat, the deactivation of the airbag should be as automated as technically reasonable and possible from a risk and homologation point of view. The goal is to minimize manual intervention while ensuring safety and compliance.

This progression from a simple advisory function to a more complex automated safety mechanism introduces new challenges and requirements in the homologation process, particularly when moving from a Lightweight (Type A) Setup to a Thorough (Type B) Setup. This case study will explore the regulatory and technical considerations involved, focusing on the implications of integrating new software-driven functionalities that directly impact critical safety components, such as airbags.

In the course of preliminary research for this case study, we identified UN Regulation 145 (UNR145) on ISOFIX in conjunction with UN Regulation 121 (UNR121) as central regulatory frameworks governing the homologation of airbags and child seats. A key requirement found in Paragraph 5.3.5 of UNR145 mandates that, if ISOFIX child restraint systems are installed in the front seat, a means of deactivating the front passenger airbag must be provided. However, there is no obligation to install ISOFIX in the front seat; it can also be installed in the rear seats only. If ISOFIX is installed in the front seat, the airbag deactivation option becomes mandatory. The regulation does not specify how this deactivation should occur, leaving room for technological flexibility. This means that airbag deactivation could theoretically be achieved using a conventional hardware switch, through the vehicle's Human-Machine Interface (HMI), or even via a mobile app.

Additionally, UNR121 requires that a tell-tale indicator in the H-pillar must illuminate if the passenger airbag is deactivated. In the context of homologation checks against UNR145, it is assumed that a manufacturer (OEM) must provide proof of compliance with Requirement 5.3.5. This likely includes documenting the methods available for airbag deactivation if multiple options are provided.

A potential concern arises when introducing a new method for deactivation through software, such as an automated system that deactivates the passenger airbag when an in-cabin camera detects a child on the front passenger seat. Such an addition could invalidate the previously approved function if this new method is not documented and reviewed during homologation for compliance with Requirement 5.3.5. This would likely necessitate a re-evaluation by a technical service, clearly indicating that the change is homologation-relevant, even if not explicitly stated in the regulations.

The reasoning behind this concern is that the introduction of a software-based function adds complexity to what has traditionally been a straightforward hardware function, significantly expanding the risk domain. For instance, manually deactivating the airbag is a deliberate action by the driver. If this is done automatically by software, it moves into the realm of assessing the function's criticality. This has significant implications, as all potential failure modes would need to be mitigated. An unintended deactivation of the airbag could have severe safety consequences for occupants, potentially leading to higher Automotive Safety Integrity Level (ASIL) requirements, redundancy considerations, and other safety measures.

Based on the findings of the homologation pre-check, it has been decided to utilize the Complex Homologation Setup for the implementation of this change request. During the detailed design phase, homologation experts are closely involved to ensure all safety and regulatory considerations are thoroughly addressed. Taking into account all the findings regarding homologation relevance and the presence of baby seats, the detailed design incorporates a mechanism where, after detecting a baby seat, the driver must explicitly confirm the deactivation of the airbag via the touch display. Following this confirmation, the airbag is automatically disabled through a newly developed API.

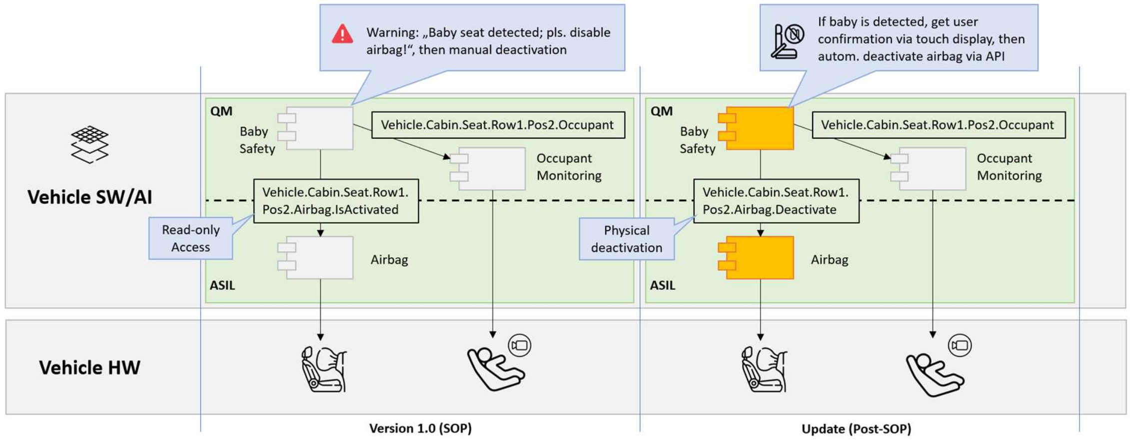

Figure 18: Airbag Deactivation via API

This new API for deactivating the airbags is an integral part of the ASIL-relevant embedded airbag control system, making it subject to rigorous safety and homologation standards. Consequently, these updates must adhere to established processes, such as ISO 26262, which governs the functional safety of electrical and electronic systems in road vehicles. Additionally, each usage of the new API for airbag deactivation will need to be thoroughly examined from a homologation perspective. Even if the calling function is a QM function, it must be ensured that using the API to disable the airbag always occurs in a safe manner, such as in combination with a manual confirmation at the HMI level. This approach is crucial for maintaining compliance with safety regulations and minimizing risks associated with automated safety-critical functions.

Use Case Comparison

In comparing the three case studies—Passenger Welcome Sequence, Acoustic Vehicle Alerting System (AVAS), and Airbags & Baby Seat Detection—we can plot them along key dimensions such as System Dependencies, Difficulty of Measurability, and Homologation Type (simple or complex).

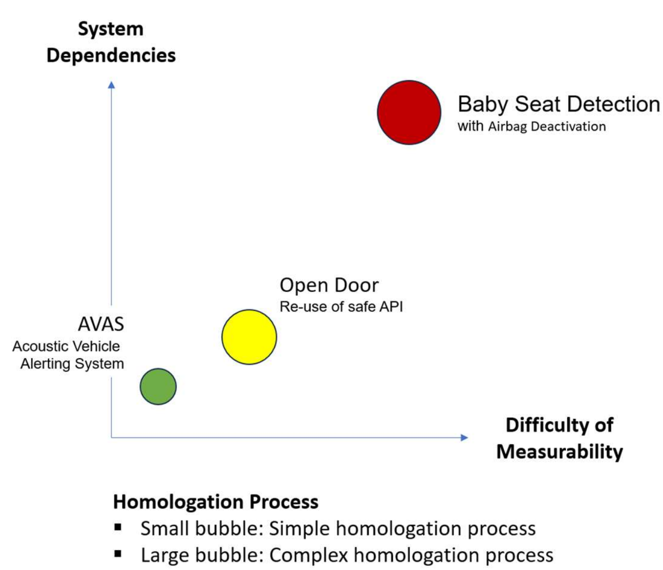

Figure 19: CoHo case studies in comparison

Here's how they compare:

- Passenger Welcome Sequence and AVAS both sit in the lower left quadrant of the grid, representing low system dependencies and simple homologation processes. Both systems are primarily focused on non-safety-critical components, with relatively straightforward testing requirements. However, Passenger Welcome Sequence may require a slightly more in-depth analysis compared to AVAS. This is because Passenger Welcome Sequence might involve interactions with more APIs, and ensuring the safe operation of all these APIs may add to the complexity. Additionally, testing the passenger welcome sequence could be slightly more complex due to its broader scope, as well as user experience elements.
- In contrast, Airbags & Baby Seat Detection is positioned significantly higher up in the upper right corner of the quadrant. This system involves high system dependencies due to the direct involvement of safety-critical components like airbags, which are governed by Automotive Safety Integrity Level (ASIL) standards. The homologation process for this system is far more complex, requiring rigorous end-to-end testing and validation to ensure that all potential failure modes are addressed. The challenge here is not only technical but also regulatory, as airbag systems must comply with stringent global safety standards. Additionally, the difficulty of measuring the change impact is much higher because airbag performance is a critical life-safety function, and every aspect of the system must be thoroughly tested in a wide range of scenarios.

Passenger Welcome Sequence and AVAS are examples of Simple Homologation processes with limited system dependencies and low measurement complexity. AVAS, in particular, can be validated through simple signal analysis (decibel levels and frequency range), while Passenger Welcome Sequence may require more API validation and integration testing, making it slightly more complex than AVAS.

Airbags & Baby Seat Detection, on the other hand, represents Complex Homologation, with high system dependencies and safety-critical implications. This requires a much more thorough and coordinated homologation process due to the ASIL compliance requirements. The complexity stems from the need to rigorously test for a wide variety of conditions and failure scenarios, with highly specific regulatory requirements that govern each aspect of the system.

This comparison highlights how the digital.auto Continuous Homologation Framework efficiently handles both simple and complex homologation processes, adapting to the specific needs of different systems depending on their safety impact, dependencies, and regulatory requirements.

Wrapping Up: Insights, Future Directions, and How to Get Involved

This paper has provided a comprehensive overview of Continuous Homologation (CoHo) for Software-Defined Vehicles (SDVs), detailing the key concepts, challenges, and practical implementation strategies for ensuring compliance throughout a vehicle's lifecycle. The framework presented by the digital.auto CoHo Special Interest Group (SIG), which includes industry leaders such as Bosch, ETAS, Certivity, TÜV Rheinland, and T-Systems, was developed based on the collective experience and expertise of its participants. The framework aims to address the complexities of maintaining regulatory compliance in SDVs, where frequent software updates and feature changes are typical.

Continuous Homologation is a dynamic and iterative process that ensures SDVs meet both Original Equipment Manufacturer (OEM) requirements and global regulations, even as new software updates or features are introduced. Unlike traditional homologation, which is conducted once prior to market release, continuous homologation integrates compliance checks at every stage of development, from design to post-production updates. This approach supports both agile development and traditional "V-model" processes, emphasizing the need for automation to manage the complexity of regulatory compliance. The use of advanced technologies such as Generative AI (GenAI) can help automate key tasks, such as generating compliance documentation, creating test scenarios, and analyzing regulatory changes, ultimately streamlining the homologation process.

Outlook

The digital.auto CoHo SIG (Special Interest Group) is planning a series of innovative projects that leverage Generative AI (GenAI) to enhance the homologation process for automotive systems. These efforts aim to streamline and automate various stages of homologation, from pre-checks to release management, ultimately improving efficiency, accuracy, and compliance in the development of safety-critical systems.

GenAI for Homologation Pre-Checks (1): The first area of focus will be the application of GenAI to support homologation pre-checks. The goal is to use AI models to automatically map requirements and global regulations, helping homologation experts quickly identify relevant standards and requirements applicable to a specific use case or system change. This will reduce manual effort, accelerate the pre-check phase, and ensure that all relevant regulatory frameworks are considered from the outset.

GenAI for Matching Requirements and Testing (2): The next step involves employing GenAI to match requirements with testing activities, including test definitions and test result analysis. By intelligently aligning test cases with specific regulatory and safety requirements, GenAI can help ensure comprehensive coverage and validation of all relevant aspects of a system. This approach will also enhance test result interpretation, enabling faster identification of compliance gaps and areas needing further testing or improvement.

GenAI for Analyzing Complex Code Bases (3): As the complexity of software in modern vehicles continues to grow, the ability to analyze large, heterogeneous code bases becomes increasingly important. The CoHo SIG aims to utilize GenAI tools specialized in analyzing such code bases to detect potential issues, inconsistencies, and vulnerabilities that may impact homologation outcomes. During the design phase, these AI-based tools can assist in identifying critical code-level dependencies and assessing the impact of code changes on functional safety. This proactive approach will help in mitigating risks early in the development process.

GenAI for Release Process Analysis (4): Finally, the CoHo SIG plans to apply GenAI during the release process to analyze changes made at the code level and determine how these changes map to original requirements, functional safety standards, and global regulations. This involves evaluating whether the implemented changes are fully compliant with all homologation requirements, identifying any deviations, and ensuring that the functional safety of the system is maintained. By automating this analysis, GenAI can help reduce the risk of errors, improve traceability, and provide a clearer understanding of the impact of software changes on regulatory compliance.

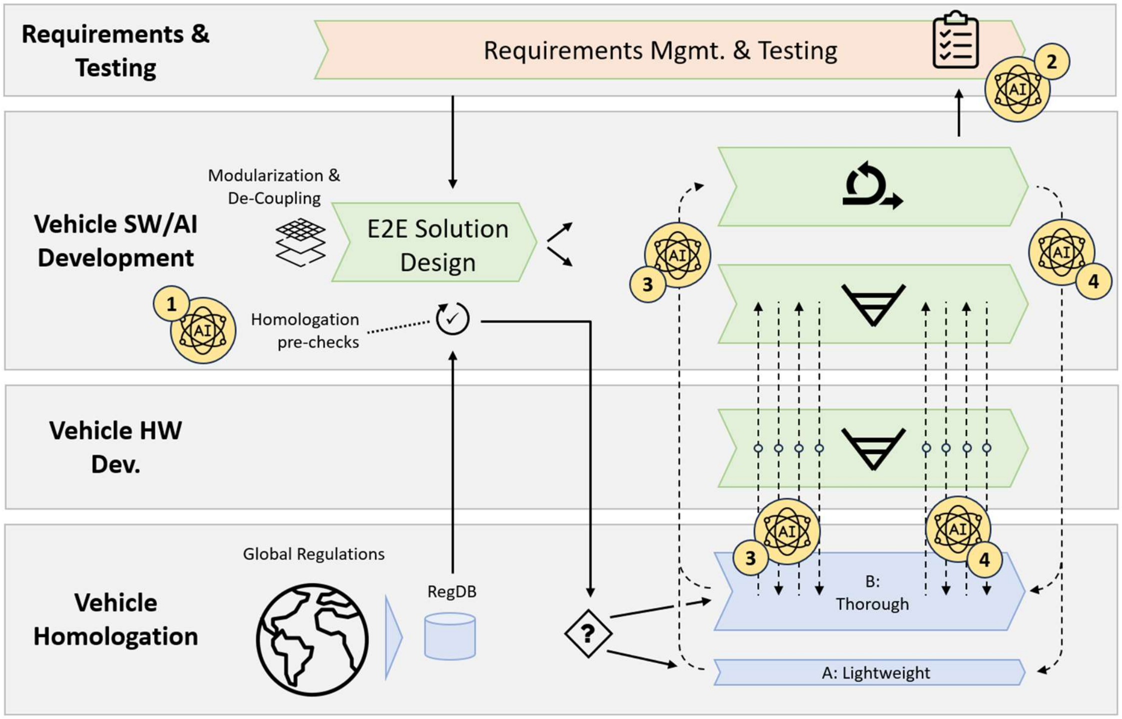

Figure 20: Utilizing GenAI to improve Continuous Homologation for SDVs

These planned advancements represent a significant step toward integrating AI-driven tools into the homologation process, paving the way for more efficient, reliable, and scalable approaches to achieving compliance in the evolving landscape of automotive software and safety systems.

In addition to GenAI, simulation and virtualization are also key topics for Continuous Homologation. Future work should look at:

- Explore simulation-based homologation pre-checks to identify compliance issues early in the development process.
- Develop methodologies for simulating real-world stress testing scenarios that are difficult or unsafe to replicate physically.
- Analyze the role of virtualization in automating compliance checks for global regulations across multiple regions.
- Assess the effectiveness of simulation tools in validating AI-driven systems
- Evaluate how virtualization can support continuous integration pipelines to enable real-time feedback and testing for updates.
- Investigate cost and time savings potential by reducing physical testing and relying more on virtual environments.

About the digital.auto CoHo SIG

The **digital.auto CoHo SIG (Special Interest Group)** is an open group of industry experts and researchers dedicated to advancing the field of **Continuous Homologation** for **Software-Defined Vehicles (SDVs)**. The group focuses on developing and applying innovative methods, frameworks, and technologies to ensure that SDVs remain compliant with regulatory standards throughout their lifecycle, despite frequent software updates and feature changes. The CoHo SIG welcomes new participants from both industry and academia who are interested in contributing to these advancements. If you want to get involved, contact **info@digital.auto** with "CoHo" in the subject line.

Currently Active Organizations:

Bosch: The Bosch Group is a leading global supplier of technology and services. Bosch Mobility brings together comprehensive expertise in vehicle technology with hardware, software, and services to offer complete mobility solutions.

ETAS: Specializes in embedded systems, providing tools, solutions, and services for developing, testing, and validating automotive software.

Certivity: Focuses on regulatory compliance management solutions, helping companies automate and streamline the homologation process.

TÜV Rheinland: A global testing, inspection, and certification organization that ensures vehicle safety, compliance, and reliability across various regulatory frameworks.

T-Systems: A leader in digital services, providing IT solutions that support connectivity, security, and data management in the automotive sector.

Research Representatives:

Ferdinand-Steinbeis-Institute: A research institute shaping the future with interdisciplinary research and practical solutions for continuous economic and societal transformation.

Hochschule Heilbronn: A university of applied sciences known for its applied research and education in automotive engineering and various other fields.

By combining the expertise of these organizations, the digital.auto CoHo SIG is at the forefront of developing practical frameworks and innovative solutions for continuous homologation in the rapidly evolving automotive industry.

How to get involved?

We are an open and collaborative community that welcomes new participants who are passionate about shaping the future of Continuous Homologation in the automotive industry. Whether you bring expertise in software, hardware, regulatory frameworks, or any related field, we encourage you to join us in driving innovation and creating best practices for the industry. We look forward to working together on exciting future projects, developing cutting-edge solutions, and sharing knowledge across the global automotive ecosystem. To get involved, simply reach out by sending an email to **info@digital.auto**—we look forward to hearing from you!

Appendix: Glossary

ASIL (Automotive Safety Integrity Level): A classification system defined by ISO 26262 that determines the required level of safety measures based on the potential risk of a system failure. ASIL ranges from A (least critical) to D (most critical).

AVAS (Acoustic Vehicle Alerting System): A safety feature for electric and hybrid vehicles that generates sound to alert pedestrians of the vehicle's presence at low speeds.

Change Request (CR): A formal proposal for modifications to a system's components or features, covering both software and hardware updates. The CR process includes analysis, design, testing, and validation before implementation.

Complex Homologation: A rigorous, end-to-end homologation process required for safety-critical components, typically involving high system dependencies and cross-disciplinary collaboration.

Continuous Homologation: A dynamic approach to managing the regulatory approval process for vehicles, allowing for ongoing updates and compliance throughout a vehicle's lifecycle.

ISO 26262: An international standard for functional safety in the automotive industry, governing the safety lifecycle and requirements for electrical and electronic systems in vehicles.

Quality Management (QM): A process and set of standards focused on ensuring that non-safety-critical components meet quality, performance, and regulatory compliance criteria.

RegDB (Regulatory Database): A centralized tool or repository, like Certivity, that organizes and provides access to global regulations, helping automakers identify and manage relevant regulatory requirements for vehicle homologation.

SDV (Software-Defined Vehicle): A vehicle whose features and functions are primarily controlled and updated through software, enabling continuous improvements and optimizations.

Simple Homologation: A streamlined homologation process applied to non-safety-critical systems or changes, requiring fewer regulatory checks and less coordination.

SUMS (Software Update Management System): A system used to manage and document all vehicle software updates, ensuring compliance, traceability, and coordination throughout the update lifecycle.