# Mastering the development challenges of safe ADAS and autonomous driving functions

April 2024

Empowering Tomorrow's Automotive Software
**www.etas.com**

# Table of contents

# Paving the way to autonomous mobility

**The automotive industry is undergoing a transformative evolution with the rapid advancement of Driver Assistance (ADAS) and Autonomous Driving (AD) technologies. These technologies are reshaping the way vehicles interact with their environment and paving the way for a future where vehicles can navigate and make decisions autonomously. In this chapter, we will explore the evolution of driver assistance and autonomous driving, delve into the classification of automation levels according to the VDA, and address the safety challenges that arise with the increasing levels of automation.**

## The evolution of driver assistance and autonomous driving

The journey towards driver assistance and autonomous driving has been marked by remarkable advancements in technology and a shift in how vehicles are perceived. From the early days of basic cruise control to the complex algorithms and sensors used in today's advanced ADAS systems, the automotive industry has witnessed a continuous evolution.

Advanced Driver Assistance Systems, often referred to as ADAS, encompass a range of features designed to aid drivers in various aspects of vehicle control and safety. These systems, such as adaptive cruise control, lane departure warning, and automatic emergency braking, have significantly enhanced the safety and convenience of driving.

Autonomous Driving (AD), the next frontier in automotive technology, takes the concept of driver assistance a step further by aiming to achieve full autonomy—where vehicles can operate independently without human intervention. AD envisions a future where cars can navigate complex environments, make split-second decisions, and communicate with each other to optimize traffic flow and safety. This evolution is not only reshaping the automotive industry but also transforming the way society interacts with transportation.

## Levels of automation: VDA classification

Central to the development of ADAS and AD systems is the classification of automation levels.

The Verband der Automobilindustrie (VDA), or the German Association of the Automotive Industry, has provided a standardized classification that categorizes the levels of automation into five distinct stages:

- **Level 1 – Assisted driving:**
  The system offers partial control over certain aspects of vehicle control, but the driver remains in control at all times.
- **Level 2 – Partial automation:**
  The system can control both direction and speed under certain conditions, but the driver must be ready to intervene at any moment.
- **Level 3 – Highly automated driving:**
  The driver relinquishes control to the system, with the expectation of resuming control when prompted by the system.
- **Level 4 – Fully automated driving:**
  The system can handle all tasks in specific scenarios, and the driver is not expected to intervene.
- **Level 5 – Full autonomy:**
  The system can operate in all scenarios without any need for driver intervention.

These levels represent a continuum of automation, with each level introducing new challenges and opportunities for development, validation, and deployment.

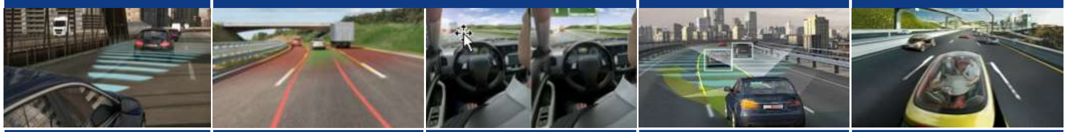
| VDA* level | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| | Assisted | Partial automation | Highly automated | Full automation | Autonomous |
| System responsibilities | Lateral OR longitudinal control in defined use-cases | Lateral AND longitudinal control in defined use-cases | | Full control for defined use-cases | Full control in ALL uses-cases |
| |  | | | | |
| Example | Adaptive cruise control | Lane keep | Highway pilot | Exit-to-exit | Shared mobility |
| Driver responsibilities | Full-time monitoring and control | Full-time monitoring and readv to resume immediate control | No monitoring. Defined period before ready to resume control | No tasks in defined use-cases | None |
| |  | | | | |

Figure 1: Automation levels according to VDA (* Verband der Automobilindustrie)

## Safety challenges with increasing automation

As automation levels increase, so do the safety challenges associated with it. While higher levels of automation promise greater convenience and reduced driver load, they also raise critical concerns related to system reliability, human-machine interaction, and fail-safe mechanisms. One of the notable shifts occurs between level 2 and level 3, where the driver transitions from being solely responsible for monitoring the environment to needing to regain situational awareness and control in a timely manner. This transition introduces a significant step-change in safety requirements from ASIL-B to ASIL-D, as the system must continue to operate flawlessly even when the driver is unable to take over due to distractions or other factors.

Furthermore, the move from assistance to automation calls for a comprehensive reassessment of traditional validation methods. Testing-by-driving, while suitable for common driving events, falls short in addressing rare and complex scenarios that can have a profound impact on system behavior.

The open-world nature of autonomous driving presents challenges in exposing systems to a diverse range of conditions and events. As the level of automation increases, the need for a robust validation approach becomes paramount to ensure the safety and effectiveness of ADAS and AD systems.

Through a comprehensive understanding of the evolution, classification, and safety considerations of ADAS and AD technologies, automotive OEMs and tier 1 suppliers can navigate the complex landscape of autonomous driving with confidence and innovation.

# The shift from assistance to automation

**In the evolution of automotive technology, the progression from driver assistance to full autonomy represents a significant paradigm shift. This chapter explores the dynamic relationship between safety and automation levels, highlights the critical role of fail-safe functionality in level 3 systems, and discusses the far-reaching impact of increasing automation on system complexity and performance.**

## The relationship between safety and automation levels

As vehicles transition from traditional manual control to higher levels of automation, a fundamental change occurs in the distribution of responsibility between the driver and the automated system. At lower levels of automation (i.e., level 1 and level 2), the driver maintains primary control of the vehicle and intervenes as necessary. However, as the automation level increases, the system assumes more responsibilities for vehicle control, leading to a reduction in driver load and engagement.

As the driver's role diminishes, the importance of functional safety amplifies. This is particularly evident when comparing level 2 systems, where the driver must be capable of immediate intervention, to level 3 systems, where the driver is expected to regain control after a defined period. The challenge lies in ensuring a seamless transition of control while addressing scenarios where the driver might be inattentive or unable to resume control promptly.

## Importance of fail-safe functionality in level 3 systems

Level 3 systems introduce a pivotal aspect: fail-safe functionality. Unlike level 2 systems, where the driver must always immediately be able to take over control, level 3 systems require the automated driving system to hand over control to the driver after a defined period. This poses a unique challenge, as the system must be equipped to handle a diverse range of scenarios, including critical situations where the driver's delayed intervention could have severe consequences.

To ensure fail-safe functionality, level 3 systems must possess a comprehensive understanding of their operational limitations and the capability to execute safe maneuvers or initiate a safe stop when faced with uncertainties. This demands a new level of reliability and redundancy to mitigate the risks associated with potential system failures or unexpected events.

## Impact on system complexity and performance

The transition from assistance to automation brings about a fundamental shift in system complexity and performance requirements. As automation levels increase, the system assumes greater responsibility for vehicle dynamics, leading to a reduction in driver load. However, this reduction in load is accompanied by a surge in the complexity of the underlying technology.

Higher automation levels necessitate more advanced algorithms, sophisticated sensor fusion techniques, and intricate decision-making processes. This surge in complexity is particularly pronounced during the leap from level 2 to level 3 and further to level 5, where the system's ability to handle a wide array of scenarios becomes paramount.

In addition to complexity, the performance demands of automated systems also escalate.

Advanced sensor arrays, high-speed data processing, and real-time decision-making capabilities are essential to achieving the level of safety and reliability expected in autonomous driving. This places substantial requirements on hardware resources such as CPUs, memory, and communication bandwidth.

In the subsequent chapters, we will delve deeper into the challenges posed by the increasing complexity and performance demands of ADAS and AD systems. By understanding the intricate interplay between automation levels, safety considerations, and system requirements, automotive OEMs and Tier-1 suppliers can chart a strategic course toward successful development and validation of autonomous driving technologies.

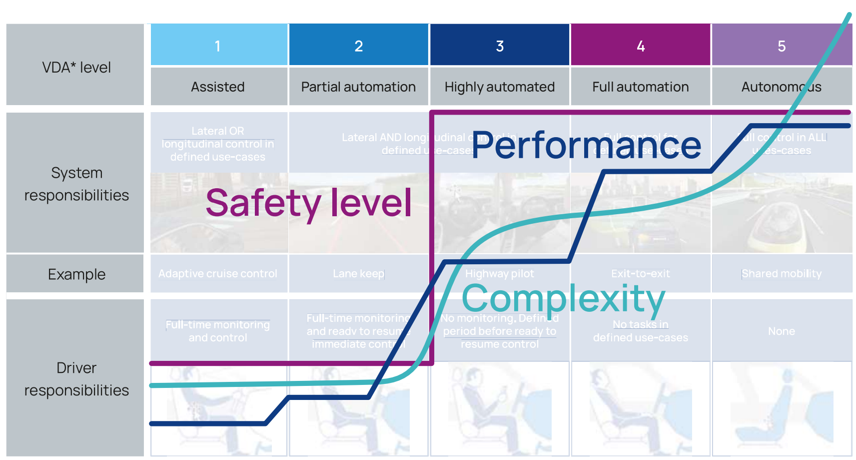



Figure 2: Safety levels driving complexity and performance requirements

(* Verband der Automobilindustrie)

# Challenges with traditional validation methods

**The development and validation of Advanced Driver Assistance Systems (ADAS) and Autonomous Driving (AD) technologies pose unique challenges that traditional validation methods struggle to address. This chapter examines the limitations and shortcomings of testing-by-driving, the significance of rare events in an open-world context, and the imperative need for capturing and exploring these rare events to ensure the safety and efficacy of ADAS and AD systems.**

## Testing-by-driving: limitations and shortcomings

Testing-by-driving has long been a cornerstone of automotive validation. Engineers would subject vehicles to a variety of scenarios, observing how software and hardware components respond. While effective for common driving events, this approach falls short when dealing with rare and unpredictable situations. In the context of ADAS and AD systems, relying solely on testing-by-driving poses several limitations:

- **Limited scenario coverage:**
  Traditional testing methods can only expose systems to a fraction of potential scenarios, leaving crucial edge cases unexplored.
- **Resource-Intensive:**
  Manually testing every conceivable scenario is resource-intensive, time-consuming, and expensive.
- **Inability to address rare events:**
  Rare but critical events, such as highly unusual traffic situations or extreme weather conditions, are difficult to encounter and test.

## Rarity of rare events and the law of big numbers

The law of big numbers suggests that even exceedingly rare events have a higher likelihood of occurrence when considered globally. Despite their rarity for any single individual, rare events can happen frequently at a global scale. This effect is illustrated in figure 3 (see below) where the required number of driving hours to experience an event increases exponentially as the probability of the event decreases.

In the realm of ADAS and AD systems, this principle is paramount. A scenario that may seem one-in-a-million to an individual driver becomes significantly more common when multiplied across a vast number of vehicles and driving conditions.

Despite their rarity for any single individual, rare events can happen frequently at a global scale. This underscores the need for comprehensive validation strategies that account for these rare yet impactful situations, ensuring that ADAS and AD systems are robust and reliable across diverse scenarios.

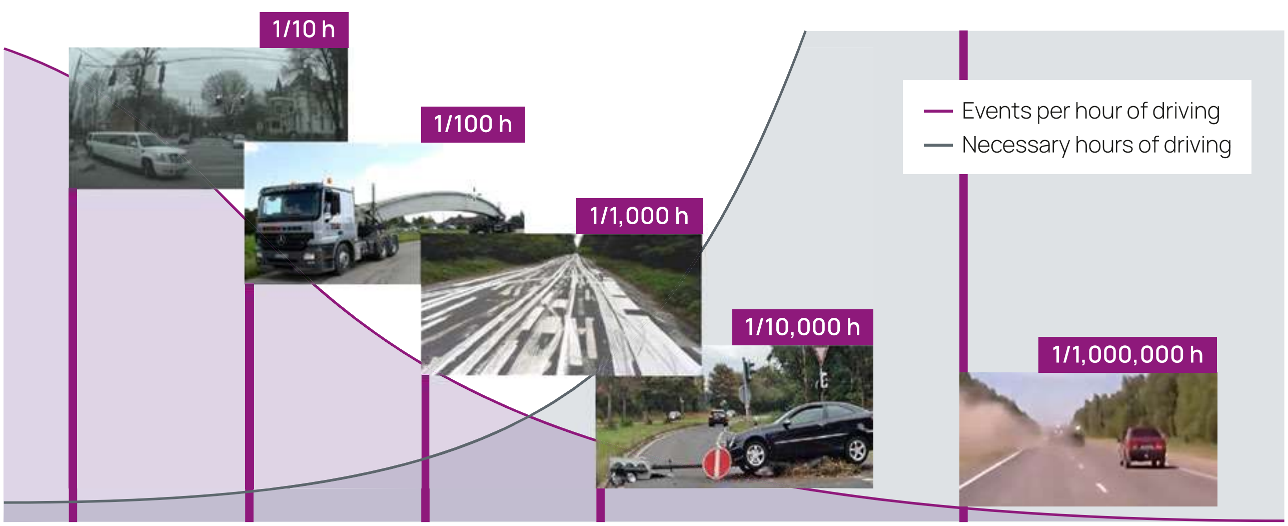



Figure 3: Rare events

## Open-world problem of autonomous driving

Autonomous driving is an open-world problem, meaning that it operates in an unconstrained environment with an infinite number of potential scenarios. While traditional testing can expose systems to common and expected events, the open-world nature of AD introduces a level of complexity that cannot be fully addressed through manual testing.

The real world is rich with uncertainties, ranging from erratic human behavior to unpredictable environmental conditions. As a result, it is virtually impossible to expose a system to every conceivable scenario through conventional testing means.

## Necessity for capturing and exploring rare events

To overcome the limitations of traditional validation methods, there is a pressing need to capture and explore rare events systematically. These rare events can include uncommon traffic situations, sensor confusion due to unusual environmental conditions, or unexpected interactions with other road users.

Capturing rare events allows developers to analyze and simulate these scenarios, gaining insights into system behavior and identifying potential vulnerabilities. The ability to reproduce and study rare events under controlled conditions is pivotal in enhancing the safety, reliability, and effectiveness of ADAS and AD systems.

By embracing a comprehensive strategy that captures and explores rare events, automotive OEMs and tier 1 suppliers can establish a robust foundation for the development and validation of ADAS and AD technologies.

# The need for successful validation

**The complexity and open-world nature of Advanced Driver Assistance Systems (ADAS) and Autonomous Driving (AD) demand innovative validation approaches that go beyond traditional testing methods. This chapter delves into the pivotal role of recomputation in validation, the challenges associated with handling rare and difficult-to-test events, and the significance of collecting validation evidence for the development of new software in the ADAS and AD domain.**

## The role of recomputation in validation

Recomputation, a cornerstone of modern validation strategies, empowers developers to systematically explore and analyze the behavior of ADAS and AD systems under various scenarios. Unlike traditional testing-by-driving, recomputation allows for the controlled and repeatable execution of specific events, both common and rare, to understand their impact on system behavior.

Through recomputation, developers can simulate the conditions that led to certain system responses, enabling them to diagnose potential issues and refine algorithms. This process not only enhances system understanding but also aids in identifying vulnerabilities and optimizing system performance.

## Handling rare and difficult-to-test events

Rare and difficult-to-test events, despite their infrequent occurrence, can have a profound impact on the safety and reliability of ADAS and AD systems. These events can range from atypical traffic situations to sensor failures in unique environmental conditions. Traditional testing methods struggle to encounter and address such scenarios comprehensively.

Recomputation serves as a powerful tool for handling rare events. By capturing data from real-world occurrences and re-executing these scenarios in a controlled environment, developers can gain valuable insights into system behavior and performance. This approach allows for systematic exploration of edge cases, contributing to a more thorough and robust validation process.

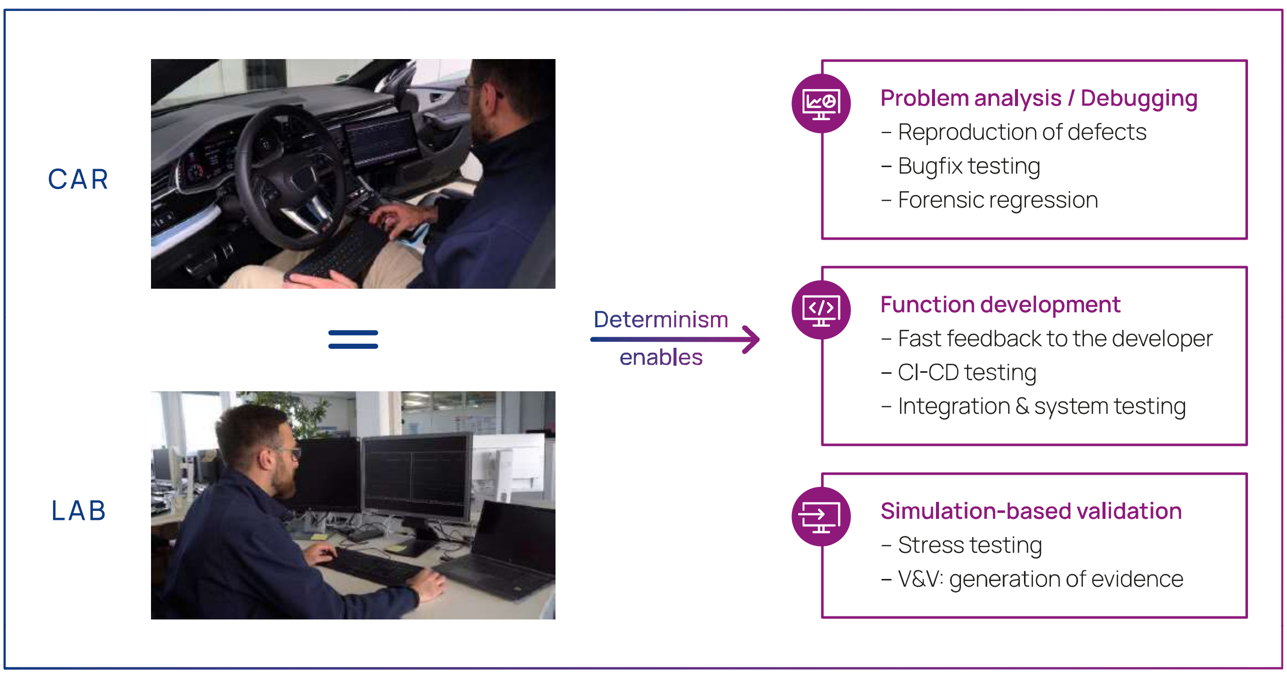



Figure 4: The role of recomputation in validation and why determinism matters

## Validation evidence collection for new software development

The development of new software for ADAS and AD systems requires a robust validation framework that generates credible evidence of system performance and safety. Recomputation plays a crucial role in establishing this framework by providing a means to validate software against a wide range of scenarios, including those that are challenging to encounter in realworld testing.

Validation evidence collection involves the compilation of data, simulations, and analyses that showcase the system's capabilities and reliability. By utilizing recomputation to recreate and analyze events, developers can compile a comprehensive repository of evidence that demonstrates the system's responsiveness, adaptability, and fail-safe mechanisms.

In the upcoming chapters, we will explore the practical implementation of recomputation and its integration into the AD Cycle. By harnessing the power of recomputation and validation evidence collection, automotive OEMs and Tier-1 suppliers can ensure the successful development and deployment of ADAS and AD technologies that meet stringent safety and performance standards.

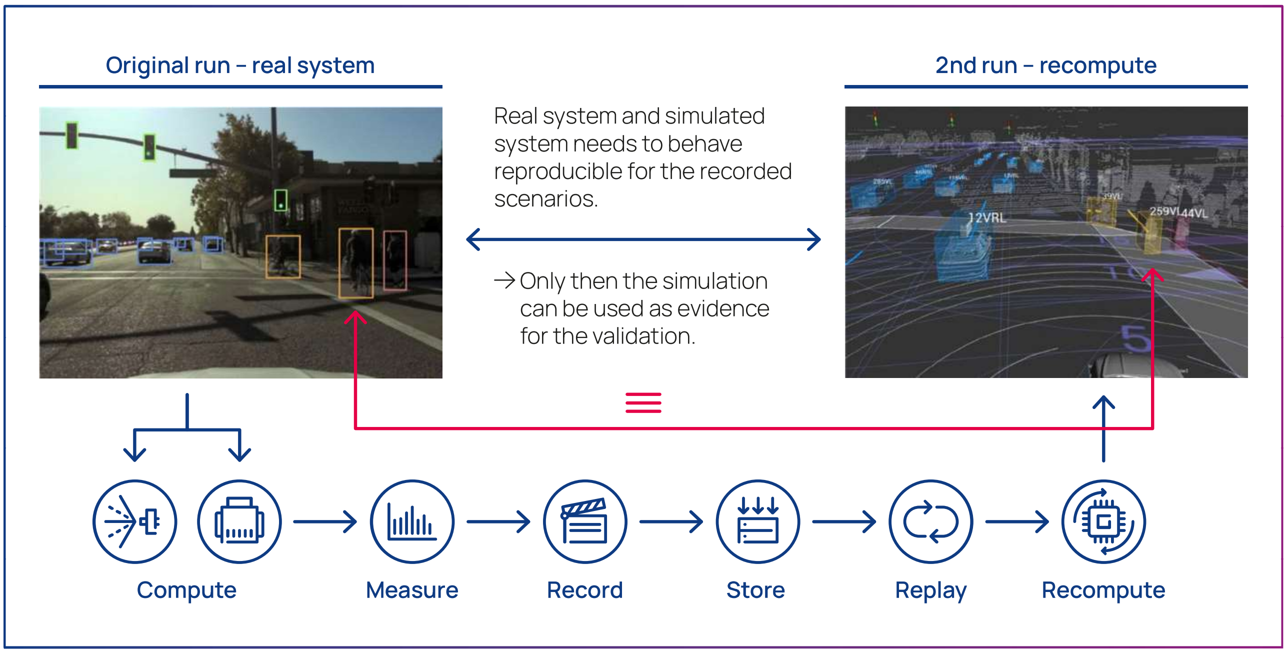



Figure 5: Open world problem and determinism

# The AD Cycle: iterative development practices

**The development of Advanced Driver Assistance Systems (ADAS) and Autonomous Driving (AD) technologies demands a structured and iterative approach that ensures thorough validation and continuous improvement. This chapter delves into the intricacies of the AD Cycle—a comprehensive framework that encompasses multiple phases—to guide automotive OEMs and tier 1 suppliers in the successful development and validation of ADAS and AD functions.**

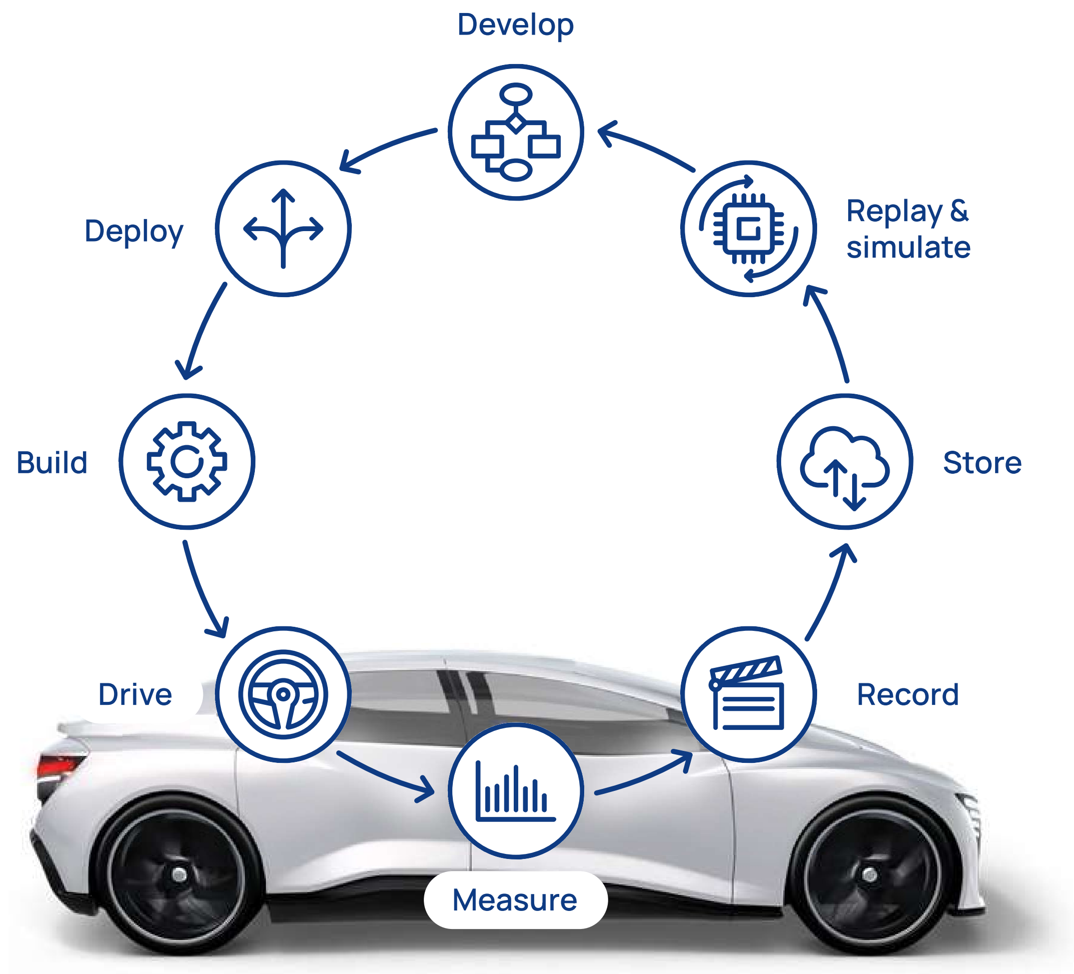



Figure 6: Full AD Cycle with real-world test drives

## Exploring the AD Cycle steps

The AD Cycle is a systematic methodology that facilitates the step-by-step development and validation of ADAS and AD functions. It encompasses five key phases, each serving a distinct purpose in the development process:

- **Design & develop:**
  Architectural design, algorithm development, and system implementation.
- **Deployment:**
  Converting design into code and configurations for deployment onto target ECUs.
- **Build:**
  Creating binary files for applications and middleware components.
- **Drive/Measure/Record:**
  Testing functions in real-world scenarios, measuring data, and recording for analysis.
- **Replay and simulate:**
  Validating and analyzing functionality using re-computation or simulation tools.

By following the AD Cycle, development teams can systematically progress through each phase, ensuring that ADAS and AD functions are rigorously tested, validated, and optimized.

## Design & develop phase: architectural design and algorithm implementation

The design & develop phase is the foundation of the AD Cycle, where the architectural design of the system takes shape. This phase involves the design and implementation of algorithms that govern the behavior of ADAS and AD functions. Engineers define the system's components, data connections, and timing requirements using a domain-specific language called YAAA (YAML-as-an-architecture).

During this phase, the system's architecture evolves, and the algorithms that govern decisionmaking and control are refined. The design & develop phase serves as the blueprint for subsequent steps in the AD Cycle.

## Deployment phase: converting design into code and bindings

In the deployment phase, the architectural design is transformed into executable code and configurations. The Code & Artifact Generator (YAAAC) automates this process, generating code bindings and configuration files for the middleware components. This phase establishes the connection between the high-level design and the tangible software that will run on target Electronic Control Units (ECUs).

By decoupling function development from deployment, the deployment phase supports software reuse and enables flexible and scalable deployment across various middleware instances.

## Drive/Measure/Record phase: testing and data collection

The drive/measure/record phase involves real-world testing of ADAS and AD functions. Built binaries are executed in test drives, during which data is measured and recorded. This phase allows developers to assess the performance and behavior of the functions in diverse driving scenarios, capturing data that will be used for validation and analysis.

The data collected during this phase serves as a critical input for the subsequent phases, providing the real-world context necessary for comprehensive validation.

## Replay and simulate phase: validation and analysis

The replay and simulate phase leverages recorded data to validate and analyze the functionality of ADAS and AD systems. This phase includes tools like Robolyzer and the RecAll Player for re-computation and simulation. Engineers can recreate scenarios, rerun functions, and analyze system behavior under controlled conditions.

The replay and simulate phase enables rigorous validation and analysis, allowing developers to explore rare and complex events that traditional testing methods struggle to address.

# Deployment of the AD Cycle

**In the pursuit of efficient and effective development practices for ADAS/AD systems, the deployment phase of the AD Cycle plays a crucial role. This phase encompasses the transition from the design and development stage to the execution and validation of the system. Within this phase, two distinct deployment approaches emerge: Test Drive and Virtual Drive.**

## Two deployment approaches: Test Drive vs. Virtual Drive

The deployment of the AD Cycle can be approached through two main strategies: Test Drive and Virtual Drive. Each approach offers unique benefits and addresses specific development needs.

### Test Drive approach

The Test Drive approach involves the traditional method of physically testing the developed ADAS/AD system in real-world driving scenarios. This approach entails deploying the system on actual vehicles and conducting real-time tests on the road. While providing valuable insights into the system's performance in a genuine environment, this approach has its limitations.

- **Real-world exposure:**
  Test Drive offers exposure to authentic driving conditions, allowing developers to observe how the system behaves in complex and dynamic situations.
- **Genuine data collection:**
  The approach enables the collection of real-time data, facilitating the analysis of system behavior and validation of algorithms.

However, the Test Drive approach presents challenges such as time constraints, logistical complexities, and limitations in encountering rare events. Nevertheless, Test Drives are indispensable to gather scenarios that reflect the true complexity and diversity of on-road experiences.

### Virtual Drive approach

The Virtual Drive approach, also referred to as the Data Shortcut approach, leverages recorded or simulated data to emulate driving scenarios. In this strategy, recorded data from real-world test drives or simulated data is used to validate and verify the ADAS/AD system's functionality.

The Virtual Drive approach offers several advantages over traditional Test Drive:

- **Efficient iteration:**
  Virtual Drive enables rapid iteration and testing, reducing the need for extensive physical test drives.
- **Rare event simulation:**
  The approach allows for the recreation of rare and difficult-to-test events, enhancing system validation.
- **Resource optimization:**
  Virtual Drive minimizes the reliance on physical vehicles and test environments, optimizing resource utilization.

Virtual Driving complements Test Drive by simulating a wide range of scenarios. While Virtual Driving can simulate numerous scenarios efficiently, Test Drives are indispensable to gather scenarios that reflect the true complexity and diversity of on-road experiences. It's the fusion of both approaches that ensures comprehensive scenario collection and system testing and ultimatively paves the way for robust ADAS/AD system development.

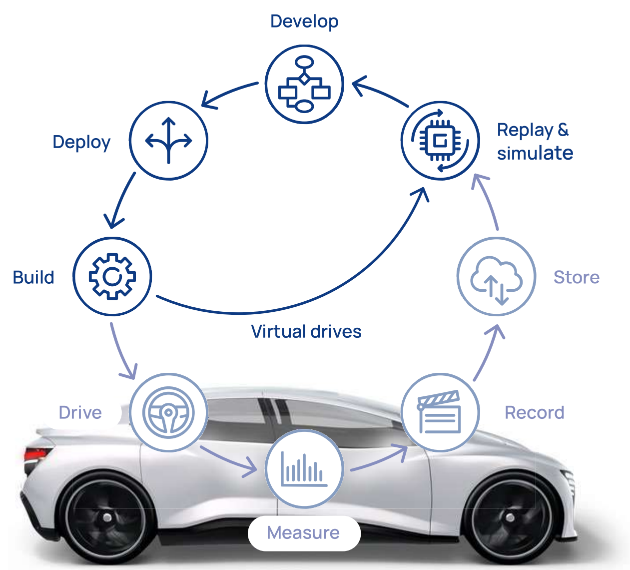



Figure 7: Fast AD Cycle with virtual drives

## Accelerating development with ETAS Deterministic Middleware Solution SDK Tools

The successful deployment of the AD Cycle, whether through Test Drive or Virtual Drive, is significantly enhanced by the suite of ETAS Deterministic Middleware Solution (EDMS) Software Development Kit (SDK) tools. These tools provide essential support for various stages of deployment, from architecture modeling to execution and analysis. While we won't delve into the specifics of individual tools, their collective impact is worth highlighting:

- **Architecture modeling:**
  The tools aid in creating a detailed representation of the ADAS/AD system's software architecture, facilitating the design and development phases.
- **Code generation and configuration:**
  EDMS SDK tools automate the generation of middlware-related code and configuration files, streamlining the deployment process.
- **Data analysis and visualization:**
  The tools offer advanced data analysis and visualization capabilities, empowering developers to gain insights into system behavior.
- **Execution and validation:**
  EDMS SDK tools support deterministic execution, allowing for reproducibility and accurate validation of the system.

Through the synergy of these tools, the AD Cycle's deployment phase becomes a robust and efficient endeavor, driving the ADAS/AD development forward.

## Comprehensive middleware solution within your ADAS/AD domain

The ETAS Middleware Solution for developing modern automated driving systems (ADAS/AD) supports SW and HW architecture modelling and high-performance communication. The solution consists of multiple tools, ECU software libraries and a robust API for the ADAS/AD development cycle. The benefits:

- Efficient development of robust driving & parking apps
- Acceleration of the ADAS/AD software development cycle for faster iterations
- Advanced debugging through forensic re-compute
- Massive reduction of physical test miles through reproducible simulation-based validation
- Seamless integration into "DevOps" environment
- Service-oriented architecture for flexibility and re-usability
- Enabling cost and time reduction for development and validation
- Support of functional safety up to and incl. ASIL-D (ISO 26262)

**Find more information online**

# How ETAS Deterministic Middleware Solution SDK Tools support the AD Cycle

**In this chapter, we explore how the suite of ETAS Deterministic Middleware Solution (EDMS) Software Development Kit (SDK) tools supports the various phases of the Autonomous Driving (AD) Cycle. These tools provide crucial functionalities that enable seamless development, deployment, execution, and validation of ADAS/AD systems.**

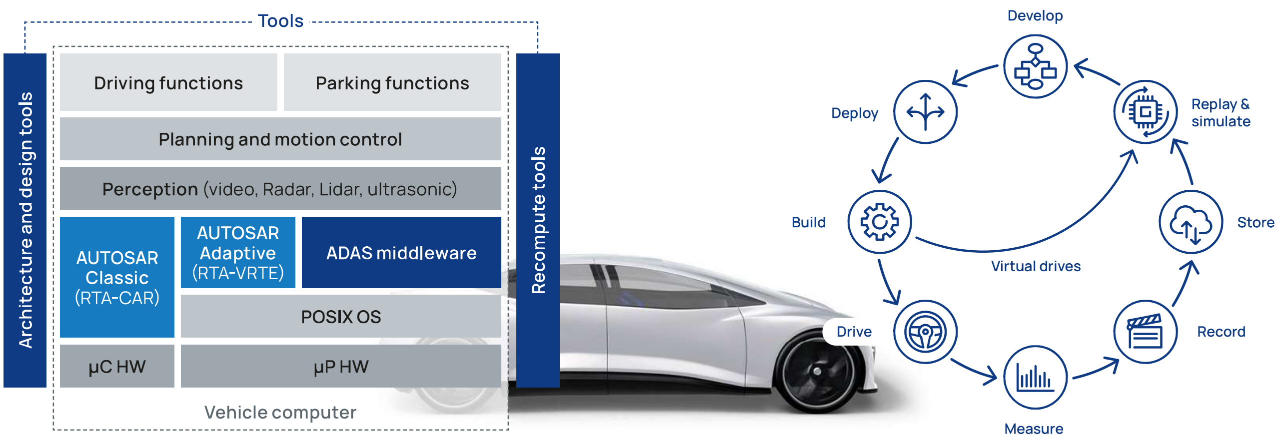



Figure 8: The Deterministic Middleware Solution - Software Development Kit and the AD Cycle

## Develop phase: designing SW architecture with YAAA

During the develop phase of the AD Cycle, the emphasis lies on creating a robust and efficient software architecture that forms the foundation of the ADAS/AD system. The EDMS SDK provides a powerful tool, the YAML As Architecture (YAAA), which assists in designing the software architecture. YAAA enables:

**Structured modeling:**
YAAA allows for the decomposition of complex systems into functional components, facilitating clear architectural design.

- **Component details:**
  YAAA captures essential details of functional components, communication interfaces, and deployment information.

- **Version control integration:**
  The use of YAAA fits seamlessly into modern Git-based workflows, promoting efficient architecture-as-code practices.

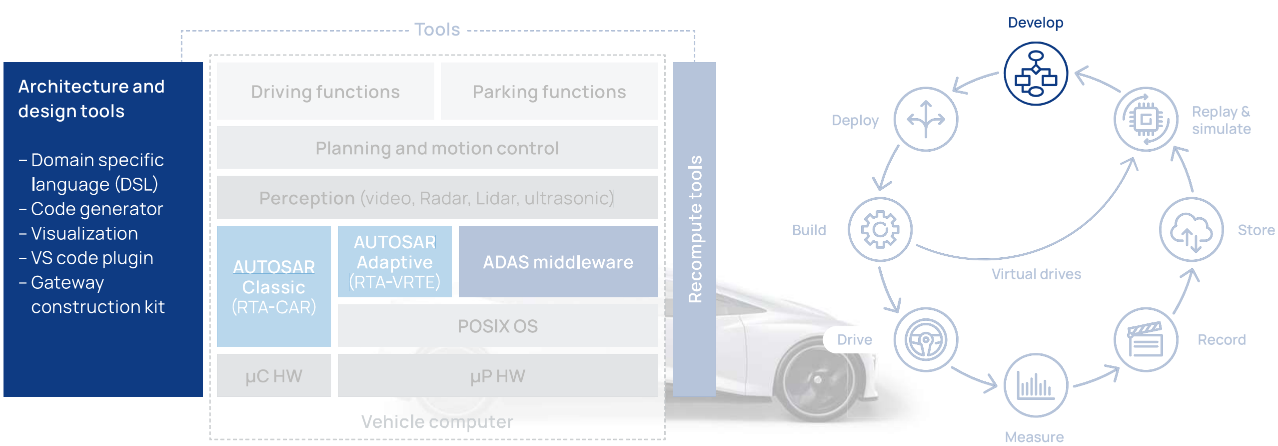



Figure 9: Develop phase

## Deploy and build phase: abstraction from hardware and OS

In the deploy and build phase, the focus shifts to converting the software architecture into executable code and preparing it for deployment. The EDMS SDK tools abstract the software from the underlying hardware and operating system, streamlining this process. The Middleware Abstraction Layer (MWALA) ensures that business logic can run across different middleware platforms that are supported by EDMS, allowing for flexible deployment.

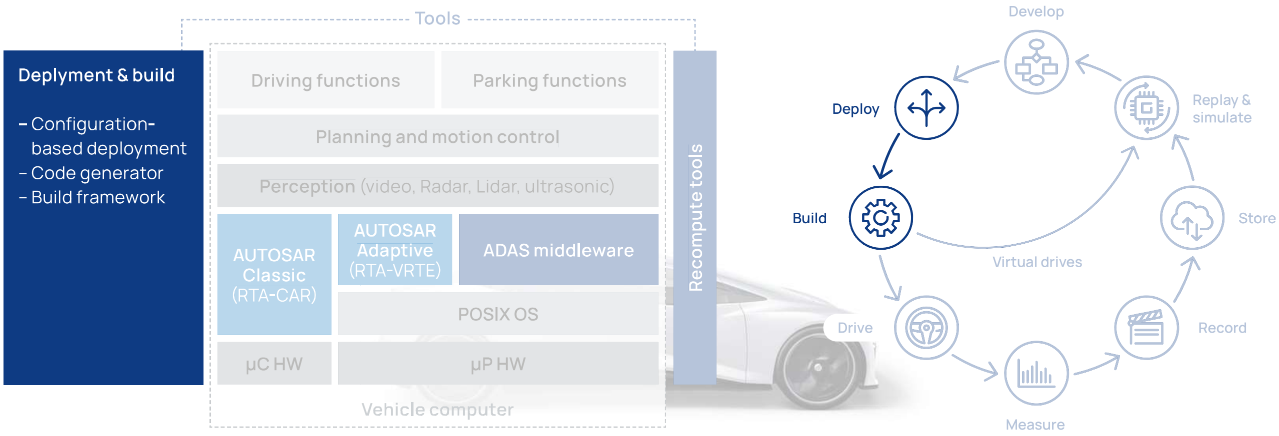



Figure 10: Deploy and build phase

## Drive phase: executing system with EDMS middleware

Executing the ADAS/AD system in real-world scenarios is a critical step in the AD Cycle. The EDMS SDK tools support this phase by seamlessly integrating with the EDMS middleware, ensuring deterministic execution. The system representation concepts within EDMS, including activities and runnables, enable reliable and consistent behavior during the drive phase.

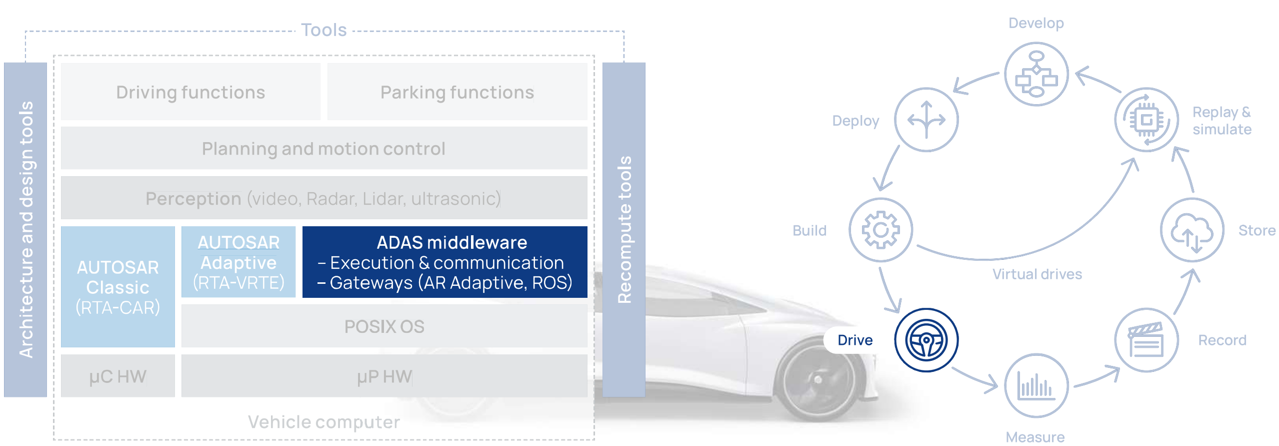



Figure 11: Drive phase

## Measure phase: recording data for reproducibility

To ensure reproducibility and effective validation, the measure phase involves recording data from system executions. The EDMS SDK tools enable the collection of accurate and detailed data, capturing execution times, activation intervals, and data flow. This recorded data becomes invaluable for subsequent analysis and validation.

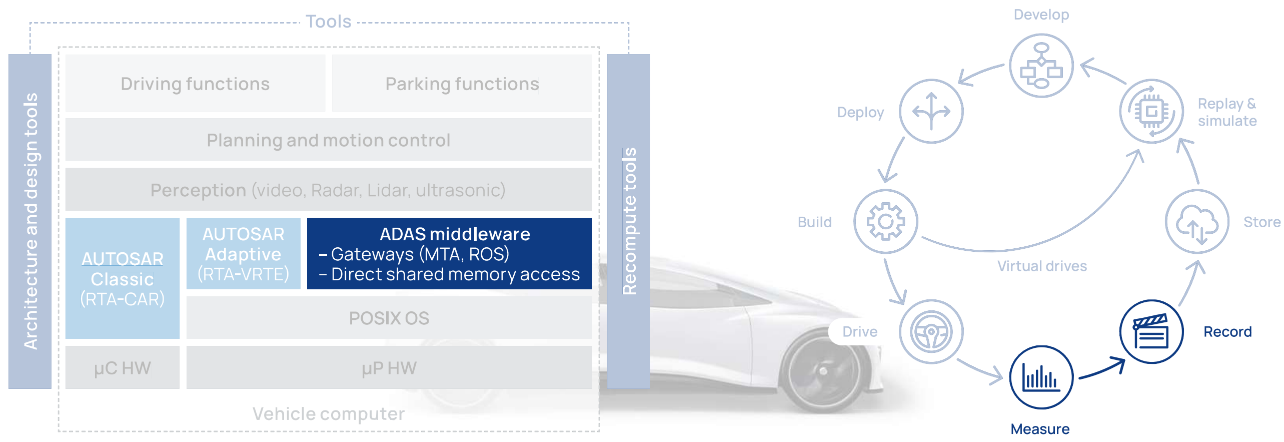



Figure 12: Measure and record phase

## Replay and simulate phase: verification and validation

The replay and simulate phase focuses on verifying and validating the ADAS/AD system's behavior using recorded data. The RecAll tool plays a crucial role in this phase. RecAll allows for the re-computation of executions, enabling forensic replay for debugging and analysis. This phase ensures that the system behaves as expected and can handle various scenarios.

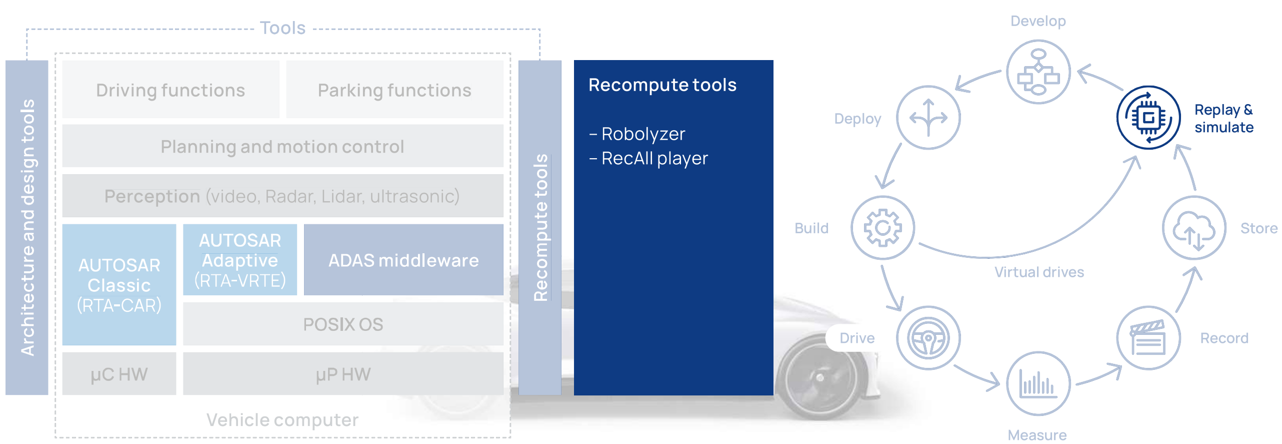



Figure 13. Replay and simulate phase

By providing comprehensive support throughout the AD Cycle, the EDMS SDK tools contribute significantly to the development and deployment of robust and reliable ADAS/AD systems.

# Benefits of ETAS Deterministic Middleware Solution

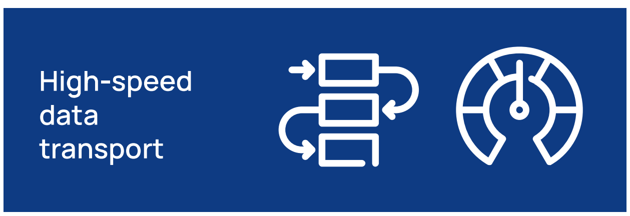
**In this final chapter, we delve into the significant benefits that the ETAS Deterministic Middleware Solution (EDMS) brings to the realm of autonomous driving. As we have explored throughout this paper, EDMS plays a pivotal role in enhancing the development, deployment, and validation of ADAS/AD systems.**


## Achieving deterministic behavior on processor-based systems

One of the most profound advantages of EDMS is its ability to achieve deterministic behavior on processor-based systems. By utilizing sophisticated scheduling and communication mechanisms, EDMS ensures that the execution of software components follows a predictable and reliable pattern. This deterministic behavior is crucial for real-time systems like ADAS/AD, where precise timing and consistent performance are paramount.


## Meeting functional safety requirements with ASIL-D support

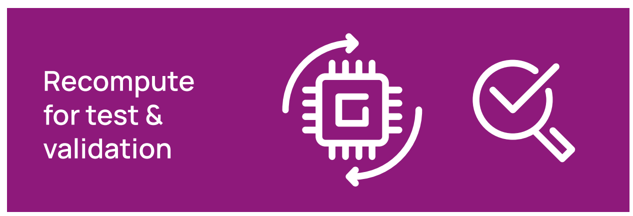
Safety is of paramount importance in autonomous driving, especially in environments where human lives are at stake. EDMS is designed with safety in mind and offers ASIL-D (Automotive Safety Integrity Level - D) support. This means that EDMS is compliant with the highest level of functional safety requirements, ensuring that ADAS/AD systems built using EDMS can meet the stringent safety standards demanded by the automotive industry.




## Handling high data rates with EDMS Middleware

Modern autonomous driving systems generate and process enormous amounts of data from various sensors and sources. EDMS Middleware, built on efficient zero-copy communication frameworks, excels at handling high data rates with low latency. This capability is essential for real-time decision-making and control, enabling

ADAS/AD systems to process and react to data in a timely manner.




## Leveraging deterministic recompute for testing and validation

A standout feature of EDMS is its deterministic recompute capability, now facilitated by the RecAll tool. This feature allows for the precise reproduction of system executions, making it invaluable for testing, debugging, and validation. Engineers can retrace and analyze specific scenarios, ensuring that the system behaves as expected and identifying any potential issues efficiently.

As we conclude our exploration of the benefits of EDMS, it becomes evident that this comprehensive middleware solution plays a pivotal role in shaping the future of Autonomous Driving. The advantages of achieving deterministic behavior, handling high data rates, meeting functional safety requirements, and leveraging deterministic recompute collectively contribute to the development of safe, reliable, and efficient ADAS/AD systems.

# Summary and conclusion

This exploration of ETAS Deterministic Middleware Solution (EDMS) has provided insights into autonomous driving development. From the evolution of driver assistance to system validation, each aspect contributes to the creation of safe and reliable ADAS/AD systems.

We began by looking at the evolution of driver assistance to full automation, classified into levels by VDA, showcasing the gradual integration of technology into vehicles. However, as automation becomes more advanced, safety challenges become more complex.

Our exploration revealed the critical interplay between safety and automation levels, emphasizing the need for fail-safe functionality in level 3 systems. This highlights the importance of design that prioritizes risk mitigation without compromise.

Traditional validation methods came under scrutiny as we uncovered their limitations, particularly in addressing rare events and the open-world problem. This underscored the necessity of capturing and exploring events that lie at the edges of possibility.

Our investigation then led us to the core of successful validation, where recomputation emerged as a powerful tool for handling challenging test cases and collecting validation evidence for new software.

The AD Cycle emerged as a recurring theme, outlining the iterative process from architectural design to validation and analysis, providing a structured framework for developing robust ADAS/AD systems.

›

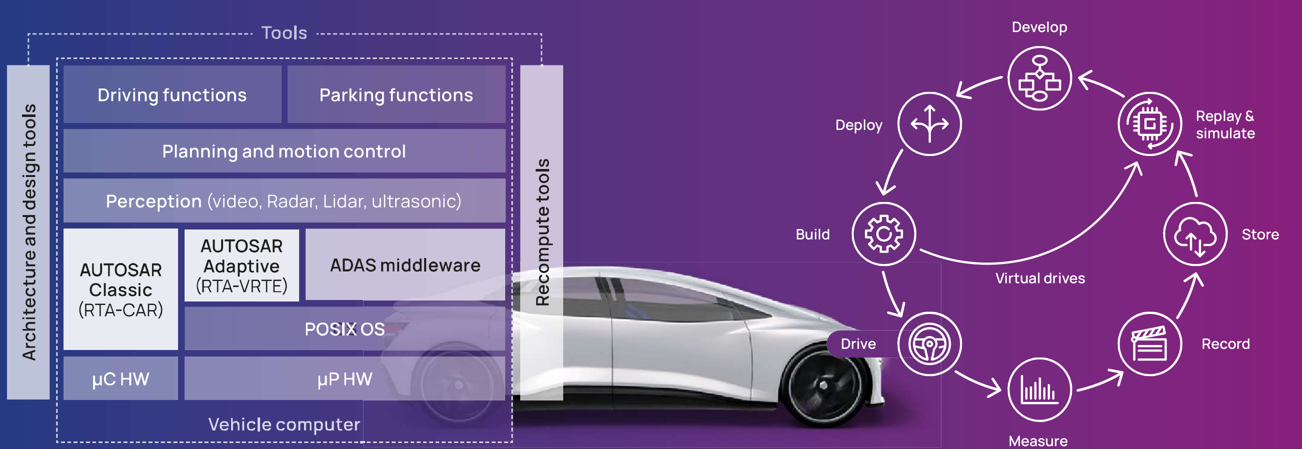



Figure 14: The ETAS Deterministic Middleware Solution – Software Development Kit and the AD Cycle

Shifting focus to deployment approaches, we contrasted Test Drives in the real world and Virtual Drives, with EDMS SDK tools playing a crucial role in streamlining abstraction and execution, bolstering the deployment phase.

We also highlighted the seamless integration of EDMS SDK tools within the AD Cycle phases, showcasing their role as trusted companions throughout the development journey of ADAS/AD systems.

The benefits of EDMS include achieving reproducible behavior from vehicles into the development environment, managing high data rates, and meeting functional safety standards. The effectiveness of deterministic recompute further strengthens this approach.

In conclusion, the future of autonomous driving relies on robust and efficient solutions like EDMS for automotive OEMs and Tier-1 Suppliers in overcoming the challenges of developing complex ADAS/AD systems. The future of autonomous driving depends on such solutions to pave the way forward.


Figure 15: Benefits of Deterministic Middleware Solution

## Ready to revolutionize your autonomous driving development?

To learn more about how ETAS Deterministic Middleware Solution can empower your organization in the era of autonomous driving, we invite you to get in touch with us. Contact our team of experts and engineers to explore how EDMS can accelerate your development, ensure safety compliance, and pave the way for the next generation of automotive innovation.

Contact us at **edms@etas.com** to schedule a consultation and discover how ETAS Deterministic Middleware Solution can transform your ADAS/AD systems.

Join us in shaping the future of autonomous driving with ETAS Deterministic Middleware Solution.

## About ETAS

Founded in 1994, ETAS GmbH is a wholly owned subsidiary of Robert Bosch GmbH with a local presence in all major automotive markets in Europe, North and South America, and Asia.

ETAS offers comprehensive solutions for the realization of software-defined vehicles in the areas of software development solutions, vehicle operating system, vehicle cloud services, data acquisition and processing solutions, integrated customer solutions and cybersecurity.

As industry pioneers in cybersecurity, we assist our customers in managing cybersecurity-related complexities, reducing cybersecurity risks, and maximizing their business potentials with a proven on- and offboard portfolio of software products and professional security services.

ETAS automotive security solutions are safeguarding millions of vehicle systems around the world – and are setting standards for the cybersecurity of software-defined vehicles.



Last updated: 04/2024

**ETAS GmbH**
Borsigstraße 24, 70469 Stuttgart, Germany
T +49 711 3423-0, info@etas.com

Are you interested in ETAS products or solutions? Please visit **www.etas.com**

Or follow us on social media: