



Enabling safe & secure modular UPdates, UPgrades and DynAmic Task reallocation and Execution for Software Defined Vehicles

D4.1 Update Procedure and Architecture V1.0

DLR

30.06.2026



Funded by
the European Union

The project, its research and all other activities leading to these results have received funding from the European Union and the granting authority CINEA under Grant Agreement No 101203060

PUBLIC

© Copyright 2025 UP2DATE4SDV Consortium Partners
101203060 — UP2DATE4SDV — HORIZON-CL5-2024-D6-01

Disclaimer:



**Funded by
the European Union**

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the granting authority (European Climate, Infrastructure and Environment Executive Agency - CINEA). Neither the European Union nor the granting authority can be held responsible for them.

All rights reserved.

Deliverable Information		
Deliverable Number	D4.1	
Deliverable Name	Update Procedure and Architecture	
Nature	Report (R)	
Dissemination Level	Public	
Contractual Date of Delivery	30.06.2026	
Actual Submission Date	30.06.2026	
Lead Beneficiary	DLR	
Lead Author	Patrick Uven	
E-Mail	patrick.uven@dlr.de	
Contributing Beneficiaries	IKL	Hebert Luyo Ane Mijangos V́ctor Raḿrez Alvaro Diez-Andino
	DLR	Patrick Uven
	BSC	Enrico Mezzetti Javier Barrera
	ITI	Joan J. Valls
	CON	Florin Anton
Reviewing Beneficiaries	IKL (WPL)	Hebert Luyo Ane Mijangos
	DLR	Patrick Uven
Work Package	4	
Related Tasks	T4.1	SASE Update Process Definition
	T4.2	SASE Reference Architecture Definition
Keywords	Software-Defined Vehicle, SDV, Horizon Europe, Automotive, CCAM, OTA, Update, Upgrade, DevOps, E/E-Architecture, Orchestration, Vehicle-to-cloud, Cloud-Edge Continuum, Reconfiguration, SASE requirements, SASE procedure, SASE architecture	

Project Information	
Contract #	101 203 060
Fixed Project Start Date	01/07/2025
Project Duration	36 months
Model Grant Agreement based on	HE MGA — Multi & Mono - V1.2
Call	HORIZON-CL5-2024-D6-01
Topic	HORIZON-CL5-2024-D6-01-01
Type of Action	HORIZON-RIA
Service	CINEA/C/03
Project Website	https://up2date4sdv.eu/

Document History		
Revision	Date	Description
V0.1	09.06.2025	Initial version
V0.2	12.06.2025	Version for review
V0.3	29.06.2025	Updated version with partners' inputs
V0.4	01.07.2026	Reviewed Version
V0.5	03.07.2026	Review comments included, version for acceptance and clearance
V1.0	03.07.2026	Final and reviewed Version

Table of contents

Table of contents.....	5
Executive Summary	7
Acronyms and Abbreviations	8
List of Figures.....	10
List of Tables.....	10
References and Background Material	10
1. Introduction	12
1.1. Project Outline	12
1.2. Technical Deliverable-Map	13
1.3. Work-Package Outline	14
1.3.1. Work-Package Objectives	14
1.3.2. Work-Package Description	14
1.3.3. Work-Package Status.....	15
1.4. Purpose and Scope of the Deliverable.....	15
2. Standards and regulations	17
2.1. Introduction	17
2.2. ISO 26262 - Functional safety	18
2.3. ISO/SAE 21434 - Cybersecurity engineering	18
2.4. UNECE R155 and R156 - Approval regulations	19
2.5. ISO 24089 - Software update engineering.....	20
3. SASE Requirements	21
3.1. Requirements Groups.....	21
4. UP2DATE4SDV reference architecture.....	22
4.1. Purpose and scope.....	22
4.2. Proposed Architecture UP2DATE4SDV	22
4.2.1. Architectural rationale: a centralised zonal E/E architecture	22
4.2.2. Architecture overview	23
4.3. Architecture Levels	24
4.3.1. Level 1 – Cloud.....	24
4.3.2. Level 2 – ECU Central Layer	25
4.3.3. Level 3 – Zonal Gateway Layer	26
4.3.4. Level 4 – Edge Devices Layer	26
4.4. Mapping ISO 24089 with UP2DATE4SDV Architecture.....	27

4.5.	UP2DATE4SDV Architecture Implementation in WP3	27
5.	Update procedure	29
5.1.	Infrastructure Process.....	29
5.1.1.	Purpose.....	29
5.1.2.	Process overview	29
5.1.3.	Safety and Security	30
5.2.	Vehicle Process	31
5.2.1.	Purpose.....	31
5.2.2.	Node roles: Bridge and recipient.....	31
5.2.3.	Process overview	32
5.2.4.	Safety and Security	33
5.3.	UP2DATE4SDV Process Implementation in WP3	35
Appendix A: Requirements:		36

Executive Summary

This document constitutes Deliverable D4.1: UP2DATE4SDV Update Procedure and Architecture [Initial]. It reports on the early results of Work Package 4 (Safety and Security Assessment). Specifically, it consolidates the outcomes of Task T4.1 (SASE Update Process Definition) and Task T4.2 (SASE Reference Architecture Definition). The deliverable has three main contributions.

First, it surveys the normative landscape for software update engineering in road vehicles, covering ISO 24089 as the central reference standard, ISO 26262 (functional safety), ISO/SAE 21434 (cybersecurity engineering), and UNECE Regulations R155 and R156 (type approval).

Second, it proposes the UP2DATE4SDV reference architecture: a centralised zonal E/E architecture structured in four levels (Cloud, Central ECU, Zonal Gateway, and Edge Device) and maps the two zones and four function families of ISO 24089 onto it.

Third, it describes the SASE-aware update procedure as a temporal projection of ISO 24089 onto both the infrastructure side (six phases) and the vehicle side (seven phases), with a transversal SASE Monitor supervising all phases. A set of sixteen safety and security requirements, derived from the normative landscape and traceable to the architecture levels, is provided as a foundational input to WP3, WP5, and the subsequent iterations of WP4 deliverables.

Acronyms and Abbreviations

Abbrev.	Full Name
AD	Autonomous-Driving
ASIL	Automotive Safety Integrity Level
CA	Consortium Agreement
CAD	Connected and Automated Driving
CCAM	Connected Cooperative and Automated Mobility
CoCo	Coordinator Contact
COTS	Commercial Off-The-Shelf
CSA	Coordination and Support Action
CSMS	Cybersecurity Management System
D	Deliverable
Diss/Expl	Dissemination/Exploitation
DoA	Description of Action
DPC	Data Privacy Coordinator
EC	European Commission
ECU	Electronic Control Unit
EU	European Union
FSIGN	Financial Signatory
GA	Grant Agreement
HAL	Hardware Abstraction Layer
HARA	Hazard Analysis and Risk Assessment
HW	Hardware
IP	Intellectual Property
JU	Joint Undertaking
LEAR	Legal Entity Appointed Representative
LSIGN	Legal Signatory (LSIGN)
M	Month
MBE	Model-Based Engineering
MBSE	Model-Based System Engineering
NDA	Non-Disclosure Agreement
NFA	National Funding Authorities
OAL	OS/MW Abstraction Layer
ODDs	Operational Design Domains
OTA	Over-The-Air
ORP	Orchestration and Reconfiguration Plane
PaCo	Participant Contact
PC	Project Coordinator
PCoCo	Primary Coordinator Contact
PGA	Project General Assembly
PPR	Periodic Project Report
PU	Public

QM	Quality Manager
R	Report
SASE	SAfety and SEcurity
SD-CCAM-V	Software-Defined CCAM vehicles
SDV	Software Defined Vehicle
SEN	Sensitive
SoM	System-on-Module
SOTA	Software-Over-The-Air
SUMS	Software Update Management System
SUP	Software Update Package
SW	Software
SyGMA	System Grant Management https://ec.europa.eu/research/participants/grants-app/reporting/DLV-101203060
TARA	Threat Analysis and Risk Assessment
TL	Task Leader
TSN	Time Sensitive Network
UCL	Use Case Leader
V&V	Verification and validation
V2C	Vehicle to Cloud
VCI	Vehicle Configuration Information
VIT	Virtual Integration Testing
VRUs	Vulnerable Road Users
WP	Work Package
WPL	Work Package Leader

List of Figures

Figure 1: Overview of the UP2DATE4SDV work-packages and their Interaction.....	13
Figure 2: Overview of the UP2DATE4SDV technical deliverable map.....	13
Figure 3: Normative landscape for software update engineering in road vehicles.	18
Figure 4: Centralised zonal E/E Architecture.....	23
Figure 5: Proposed Architecture UP2DATE4SDV.....	24
Figure 6. Cloud Sub-Components.....	24
Figure 7. ECU Central Sub-components.	25
Figure 8. Zonal Gateway Sub-Components.	26
Figure 9. Edge Devices Sub-components.	26
Figure 10: Architecture of the ORP[1].....	28
Figure 11. End-to-end Infrastructure update process.....	30
Figure 12. SASE Monitor as a transversal supervision layer over the six phases of the infrastructure update process.	30
Figure 13. Bridge and recipient roles depending on the campaign target.	32
Figure 14. Vehicle Process based on ISO 24089.....	33
Figure 15. SASE Monitor over the vehicle update process.	34

List of Tables

Table 1:SASE Requirements groups according to the ISO 24089 function families.....	21
Table 2: ISO 24089 to UP2DATE4SDV Architecture Mapping	27
Table 3: Mapping between the infrastructure update phases and the SASE requirements.	30
Table 4: Mapping between the vehicle update phases and the SASE requirements.	34

References and Background Material

Internal Background

- [1] D3.1 - Requirements & Drafts of the Orchestration and Reconfiguration Plane (ORP)
- [2] D3.2 - DevOps and V&V Framework [Initial]
- [3] D3.3 - Safe and Secure Orchestration and Reconfiguration Plane [Initial]
- [4] D4.2 - SASE Mechanisms [Initial]
- [5] D5.1 - Demonstrator Definitions, Metrics, and Plan

External Background

- [6] UP2DATE Consortium, "D3.2 – Architecture definition," v02.00, H2020 UP2DATE Project, Oct. 2021. [Online]. Available: https://h2020up2date.eu/wp-content/uploads/2021/10/D3.2_ArchDefinition_v02.00.pdf
- [7] Road vehicles — Functional safety, ISO 26262, International Organization for Standardization, Geneva, Switzerland, 2018. [Online]. Available: <https://www.iso.org>
- [8] Road vehicles — Cybersecurity engineering, ISO/SAE 21434:2021, International Organization for Standardization / SAE International, Geneva, Switzerland, 2021. [Online]. Available: <https://www.iso.org/standard/70918.html>
- [9] Road vehicles — Software update engineering, ISO 24089:2023, International Organization for Standardization, Geneva, Switzerland, 2023. [Online]. Available: <https://www.iso.org/standard/77796.html>
- [10] Uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system, UN Regulation No. 155, United Nations Economic Commission for Europe, Geneva, Switzerland, 2021. [Online]. Available: <https://unece.org/transport/documents/2021/03/standards/un-regulation-no-155-cyber-security-and-cyber-security>
- [11] Uniform provisions concerning the approval of vehicles with regards to software update and software updates management system, UN Regulation No. 156, United Nations Economic Commission for Europe, Geneva, Switzerland, 2021. [Online]. Available: <https://unece.org/transport/documents/2021/03/standards/un-regulation-no-156-software-update-and-software-update>
- [12] Functional safety of electrical/electronic/programmable electronic safety-related systems — Parts 1–7, IEC 61508:2010, International Electrotechnical Commission, Geneva, Switzerland, 2010. [Online]. Available: <https://www.iec.ch>
- [13] Industrial communication networks — Network and system security — Part 3-3: System security requirements and security levels, IEC 62443-3-3:2013, International Electrotechnical Commission, Geneva, Switzerland, 2013. [Online]. Available: <https://www.iec.ch>
- [14] Security for industrial automation and control systems — Part 4-2: Technical security requirements for IACS components, IEC 62443-4-2:2019, International Electrotechnical Commission, Geneva, Switzerland, 2019. [Online]. Available: <https://www.iec.ch>
- [15] NXP Semiconductors, How Zonal E/E Architectures with Ethernet Are enabling software-defined vehicles, 2023. [Online]. Available: <https://www.nxp.com/company/about-nxp/smarter-world-blog/BL-HOW-ZONAL-EE-ARCHITECTURES>

1. Introduction

The European research project UP2DATE4SDV (Enabling safe & secure modular UPdates, UPgrades and DynAmic Task reallocation and Execution for Software-Defined Vehicles), aims to develop a comprehensive ecosystem for seamless and efficient software updates, hardware upgrades and situation-dependent reconfigurations of software-defined vehicles.

1.1. Project Outline

The project focuses on the definition and development of three abstraction layers

- the hardware abstraction layer (HAL)
- the OS/MW abstraction layer (OAL) for the operating system (OS) and middleware (MW)
- as well as on the exploration and prototyping of a secure orchestration and reconfiguration layer (ORP) between the vehicle and the cloud.

This will improve the reliability and performance of the CCAM system, extend vehicle life and reduce electronic waste. In addition, the project's new modularity concepts will accelerate the transition to safer, more sustainable automated driving and enable incremental steps towards Level 4 automation and longer vehicle lifecycles.

The overall UP2DATE4SDV goal is to create a comprehensive ecosystem for updatable, upgradable, and reconfigurable software-defined connected and automated vehicles. This will be achieved by developing essential abstraction layers, advanced orchestrators, and an automated robust realization environment, all while adhering to the highest safety and cybersecurity standards. To achieve this vision, the project will target the following objectives:

- O1. Creation of a reconfigurable cloud connected zonal vehicle architecture blueprint based on heterogeneous COTS HW (exploiting and building on a unified HAL)**
- O2. Establishing a resource composable SW deployment, execution and communication reference layer, on top of existing open-source hypervisor and OS stacks (exploiting and building on a unified OAL)**
- O3. Realization of a dynamic update, orchestration & reconfiguration plane**
- O4. Provide harmonized and simplified development, integration and validation processes and framework for SDV applications**
- O5. Assess joint Safety and Security (SASE) of all elements from O1, O2 and O3**
- O6. Demonstration of UP2DATE4SDV solution on several CCAM relevant Use Cases**
- O7. Active participation around CCAM and SDV Initiative community and ecosystem**

To this goal, the demonstrators and use-cases (developed in WP5) define requirements (pink arrows) for the HW and the HAL (developed in WP1), the SW and the OAL (developed in WP2) as well as for the ORP (developed in WP3). Based on that, WP1, WP2 and WP3 provide a basic functional HW-SW-demonstrator for the use-cases – UC1- after Phase 1 (M6), UC2, UC3, and UC4 after Phase 2 (M24), and finally UC5 and UC6 after Phase 3 (M36).

Besides the pure communication of results (green arrows) the proper integration of the results of WP1, WP2, WP3, and WP4 within the use-cases depends on three further cooperation aspects: the safety and security assessment (yellow arrows), the update-, upgrade-, and reconfiguration-processes and the involved component-interfaces (purple arrows), and the cloud-based MBSE-representation of

actual HW-/SW-components during the DevOps cycle establishing the real-world to cloud connection according to digital twins (blue arrows). Finally, while WP6 oversees the dissemination, communication, and exploitation of the project's results, WP7 coordinates all the WP and their communication, internally as well as externally to the SDV-Initiative and to other CCAM-partnerships. Figure 1 summarizes the described interaction and communication between the work-packages (WPs) of UP2DATE4SDV.

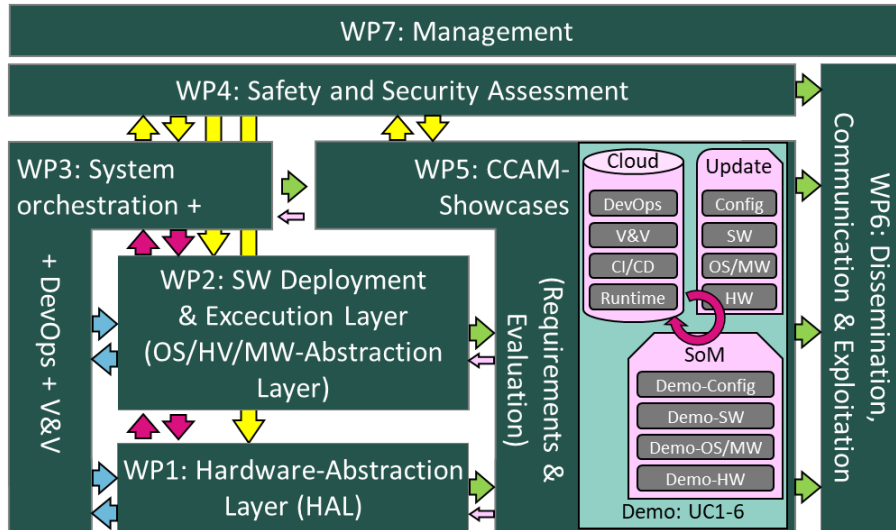


Figure 1: Overview of the UP2DATE4SDV work-packages and their Interaction.

1.2. Technical Deliverable-Map

UP2DATE4SDV has defined 38 deliverables of which 10 have an organizational nature and 28 a technical nature. To provide orientation to the reader, Figure 2 shows a technical deliverable map that gives a structural overview of all technical deliverables and their content-wise relation.

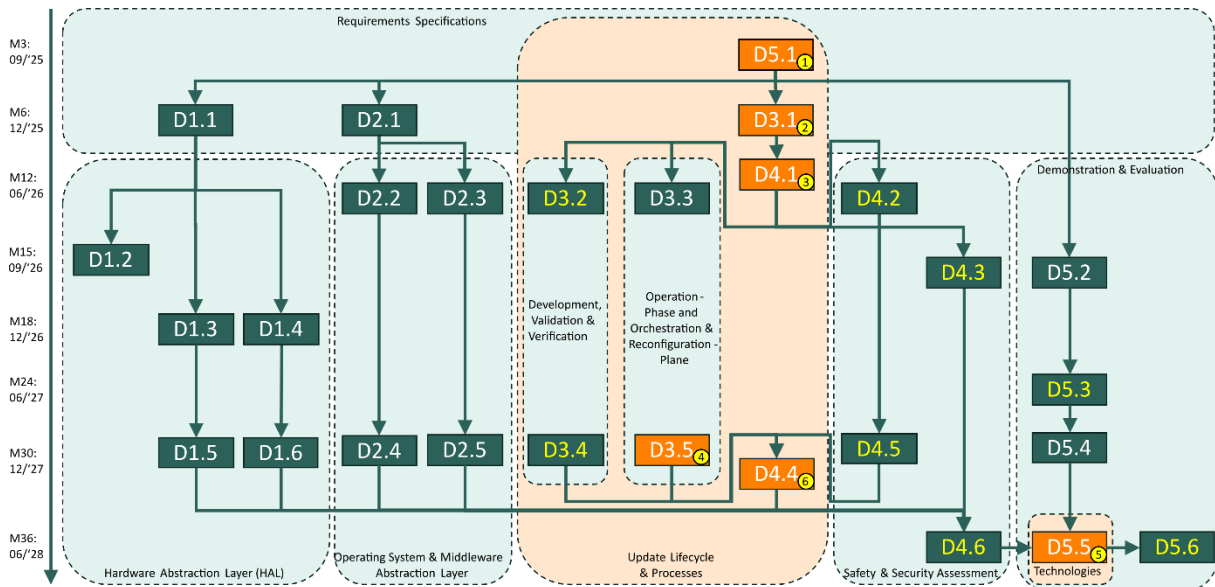


Figure 2: Overview of the UP2DATE4SDV technical deliverable map, showing the deliverables of the technical work-packages WP1-WP5, with restricted deliverables highlighted by the yellow text, a main-track highlighted by the orange nodes, and a recommended reading-order in the yellow circles. The arrows between deliverables are indicating the content development.

Deliverables are shown as rectangles, which are connected by arrows if the content of one deliverable is based on the contents of previous ones. The text in the rectangles shows the deliverable number and it is highlighted in yellow, in case of restricted deliverables, whereas public deliverables are labelled in white. The light-green and light-orange areas form technical content-categories, close but not equal to the work-packages. They show the hardware abstraction layer (HAL) in WP1, the Operating-System and Middleware Abstraction-Layer (OAL) in WP2, the Safety and Security Assessment in WP4, and the Demonstrators and the Evaluation in WP5, but they combine the Orchestration and Reconfiguration-Plane of WP3 with the safety- and security-analysis and corresponding measurements from WP4, forming the area of the Update Lifecycle and Processes, where Development, Validation and Verification, and Orchestration and Deployment of Updates are covered. Furthermore, the four initial deliverables of the work-packages requirements-definitions are also combined in a separate area at the very top. To get an overall impression of the full project, the main story-track of deliverables is highlighted in orange, and a suggested reading-order is indicated by numbers in yellow circles.

1.3. Work-Package Outline

1.3.1. Work-Package Objectives

WP4 provides the safety and security (SASE) foundations of the UP2DATE4SDV project. Its overarching objective is to ensure that UP2DATE4SDV results are aligned, designed and verified in compliance with the functional safety and cybersecurity standards applicable to software-defined vehicles (SDV).

To achieve this, WP4 pursues four complementary goals:

- (i) define a combined SASE procedure for managing and executing software updates and dynamic reconfigurations of any system component, integrating functional safety and cybersecurity concerns end-to-end;
- (ii) develop a robust reference zonal vehicle architecture pattern that provides the required means to update and dynamically reconfigure any system component while preserving safety and security;
- (iii) implement preventive and proactive SASE mechanisms as configurable open-source libraries, ensuring compliance of SDV applications with ISO 26262 and ISO/SAE 21434;
- (iv) and continuously assess the proposed concepts in collaboration with external certification authorities (TÜV Rheinland) and relevant automotive stakeholders, maintaining alignment with current and emerging safety and security standards and regulations.

1.3.2. Work-Package Description

WP4 is structured into four tasks, addressing the SASE dimension of the UP2DATE4SDV update and reconfiguration ecosystem.

Task T4.1 SASE Update Process Definition defines a comprehensive and integrated procedure for managing and executing updates and dynamic reconfigurations of any system component, maintaining both safety and security under any operational scenario, including roll-out and roll-back situations. The task identifies, jointly with WP3, the infrastructure and tooling required to automate update and upgrade processes while preserving functional and non-functional properties.

Task T4.2 SASE Reference Architecture Definition defines a modular and scalable reference zonal vehicle architecture providing the architectural components (orchestrator, update manager, online/offline monitor, compatibility checker...) and the safety and security mechanisms (diagnostics, encryption, authentication, intrusion detection, probabilistic preventive risk evaluators...) to be integrated into the UP2DATE4SDV platform, ensuring compliance with relevant European and international standards and regulations.

Task T4.3 SASE Mechanisms Implementation implements the preventive and proactive SASE mechanisms identified in T4.2 as configurable open-source libraries, considering requirements from functional safety, cybersecurity, and regulatory standards applicable to automotive SDV applications.

Task T4.4 — Continuous Assessment performs ongoing assessment of all WP4 concepts from a safety and security perspective, elaborating a safety and security case for the UP2DATE4SDV architecture instantiated on a CCAM showcase at a specific ASIL and Security Level. The task coordinates the assessment activities with external certification authority TÜV Rheinland through defined milestones (kick-off, documentation review, assessment meetings).

1.3.3. Work-Package Status

WP4 activities started as planned in October 2025 (M4). At the time of writing (M12), all four tasks are active. Collaboration among WP4 partners has been maintained through regular WP calls and dedicated task-level coordination sessions. Requirements, architecture decisions, and SASE mechanisms have been collected and consolidated through the project collaboration platform.

This deliverable, together with D4.2 (SASE Mechanisms [Initial], due M12) and D4.3 (SASE Assessment [Initial], due M15), constitutes the first set of WP4 outcomes. The SASE requirements listed in Appendix A: Requirements: provide the foundational engineering input for the project, and the final iterations of the WP4 deliverables (D4.4 and D4.5 at M30, D4.6 at M36). While these requirements can benefit from integration and refinement along the course of the project, WP4 progresses as planned and currently no delays are expected.

1.4. Purpose and Scope of the Deliverable

Deliverable D4.1 belongs to the initial set of WP4 deliverables due at M12 and represents the first formal consolidation of the safety and security framework governing software update and reconfiguration operations in UP2DATE4SDV. Together with D4.2 (SASE Mechanisms [Initial]) and D4.3 (SASE Assessment [Initial], due M15), it provides the foundational SASE baseline on which the full WP4 assessment will be built over the course of the project.

The deliverable serves three primary purposes. First, it establishes the normative context by analyzing the applicable standards and regulations: ISO 24089, ISO 26262, ISO/SAE 21434, UNECE R155 and R156 and making explicit the requirements they impose on any software update engineering system for road vehicles. Second, it proposes the first iteration of the UP2DATE4SDV reference architecture and maps it to the two-zone and four-function-family model of ISO 24089. Third, it documents the SASE-aware update procedure on the infrastructure and vehicle sides, with an explicit transversal SASE supervision layer. The set of SASE requirements in Appendix A: Requirements:, traceable to both normative sources and architecture levels, is provided as a concrete engineering artefact for cross-WP coordination.

The scope of this first iteration covers the initial results of T4.1 and the early outcomes of T4.2. SASE mechanisms (T4.3) and the continuous assessment (T4.4) are reflected here at the level of

requirements and architectural provisions only; their detailed treatment will be provided in D4.2, D4.3, and the final deliverable iterations.

2. Standards and regulations

2.1. Introduction

Software-defined vehicles are increasingly designed to support software updates and hardware upgrades throughout their entire operational lifetime. While Over-The-Air (OTA) update capability delivers clear benefits as extending vehicle functionality, correcting defects, and maintaining security posture after deployment, it also introduces risks that conventional automotive development frameworks were not originally designed to address. An update operation modifies software that may be safety-critical, traverses communication channels exposed to adversarial threats, and must be managed in a way that preserves type-approval compliance across the vehicle's service life. These three dimensions functional safety, cybersecurity, and regulatory compliance define the normative landscape that governs software update engineering in road vehicles and that this chapter describes.

Three standards structure this space. ISO 26262 establishes the functional safety framework for road vehicle electrical and electronic systems, providing the safety integrity levels, hazard analysis methods, and development processes that must be upheld whenever an update affects safety-relevant functions. ISO/SAE 21434 establishes the cybersecurity engineering framework, covering threat analysis and risk assessment, security controls, and the governance of cybersecurity risks across the full vehicle lifecycle, including the update channel and its associated infrastructure. ISO 24089, published in 2023, addresses software update engineering as a discipline in its own right: it defines the organisational processes, infrastructure capabilities, vehicle-side functions, and operational procedures that together constitute a Software Update Management System (SUMS). ISO 24089 sits at the centre of this landscape. It explicitly requires conformance with ISO 26262 and ISO/SAE 21434 wherever update operations affect safety-related or cybersecurity-relevant functions, making it the normative integration point for the other two standards.

Completing the landscape are the two UNECE regulations adopted by Working Party 29 (WP.29) in 2021. UNECE R155 and R156 are not technical standards but legal instruments: they define what vehicle manufacturers must demonstrate to obtain and maintain type approval in the contracting parties to the 1958 Agreement. R155 requires a certified Cybersecurity Management System (CSMS) and R156 a certified Software Update Management System (SUMS). Both regulations explicitly recognise the corresponding technical standards ISO/SAE 21434 for R155 and ISO 24089 for R156 as the primary evidence of compliance. For any vehicle deployed in jurisdictions where these regulations apply, the standards described in this chapter are therefore not merely engineering best practice but effectively a prerequisite for market access. Figure 3 illustrates the full normative landscape and the relationships between its elements.

The remainder of this chapter describes each standard individually, with the level of detail necessary to understand how UP2DATE4SDV builds on them. ISO 24089, as the structural reference for WP4 and the principal organising framework for the update procedure described is described last and in the greatest depth.

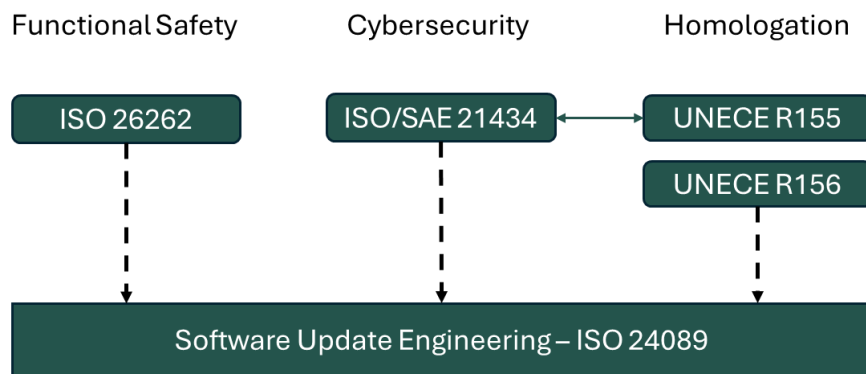


Figure 3: Normative landscape for software update engineering in road vehicles.

2.2. ISO 26262 - Functional safety

ISO 26262, *Road vehicles - Functional safety*, is the automotive instantiation of IEC 61508. It applies to safety-related electrical and electronic systems in production passenger cars and provides a complete framework for managing functional safety throughout the system lifecycle: from concept and hazard analysis through development, production, operation and decommissioning. The standard is published in twelve parts (ISO 26262-1 through ISO 26262-12), each addressing a specific lifecycle stage or supporting topic.

The central concept of ISO 26262 is the Automotive Safety Integrity Level (ASIL), derived from a Hazard Analysis and Risk Assessment (HARA) of the item under consideration. ASILs range from QM (quality management, no safety relevance) to ASIL D (most stringent), and they drive the rigour required at every later stage: the safety mechanisms to implement, the verification and validation effort, the documentation, and the independence of the assessors.

For software update engineering, ISO 24089 explicitly mandates conformance to ISO 26262-6 (Product development at the software level) and ISO 26262-8 (Supporting processes) wherever safety-related functions are affected by an update. Other parts of the standard are also relevant: ISO 26262-3 (Concept Phase) for the HARA underlying any safety-related decision; ISO 26262-4 (Product development at the system level) for system-level integration and validation; and ISO 26262-5 (Product development at the hardware level) and 26262-11 (Guidelines on application of ISO 26262 to semiconductors) for hardware-level safety mechanisms, particularly relevant for the hardware safety monitors and non-volatile memory protections discussed later in this deliverable.

Within UP2DATE4SDV, ISO 26262 anchors every requirement and mechanism that addresses functional safety during update operations: maintenance of safe vehicle state, hardware-level monitoring during installation and activation, post-update integration validation, and corrective actions in the event of a failed update.

2.3. ISO/SAE 21434 - Cybersecurity engineering

ISO/SAE 21434, *Road vehicles - Cybersecurity engineering*, is the automotive cybersecurity counterpart to ISO 26262, jointly developed by ISO and SAE and published in August 2021. It provides a framework for managing cybersecurity risks across the entire lifecycle of road vehicle electrical and electronic systems, from concept through development, production, operation, maintenance and decommissioning.

The standard is organised around several core ideas. First, it requires the organisation to operate a Cybersecurity Management System (CSMS) covering policies, processes, roles and supporting activities (Clauses 5 to 7). The second is the Threat Analysis and Risk Assessment (TARA), which is the systematic method for identifying threats, evaluating risks and deriving cybersecurity goals (Clauses 9 and 15). The third, it structures the engineering activities along the concept-to-validation flow with cybersecurity goals, requirements, design, integration and verification (Clauses 9 to 11).

The fourth is the set of continual activities that extend cybersecurity governance beyond initial release, covering production, operations and maintenance, and end of cybersecurity support (Clauses 12 to 14), together with vulnerability management and incident response as recurring duties.

For software update engineering, ISO 24089 explicitly mandates conformance to ISO/SAE 21434 wherever update operations affect cybersecurity-relevant assets, which in practice applies to virtually all update operations: the package itself, its signing keys, the communication channels, and the vehicle configuration information are all cybersecurity assets in the sense of the standard. Several specific work products of ISO/SAE 21434 are reused directly within UP2DATE4SDV: TARA outputs, cybersecurity goals, cybersecurity case, and vulnerability management process.

ISO/SAE 21434 also has a regulatory dimension: it is recognised by UNECE R155 as evidence of compliance with the requirements for a Cybersecurity Management System, which makes it the standard route to vehicle approval in jurisdictions that have adopted R155.

2.4. UNECE R155 and R156 - Approval regulations

The United Nations Economic Commission for Europe (UNECE) issues the regulations that govern vehicle approval in most of Europe and in several other countries. Unlike ISO standards, these regulations are legally binding: manufacturers must comply with them to sell vehicles in those markets. Two of them, both adopted in 2020, directly govern cybersecurity and software updates. These regulations are not technical standards in the ISO sense, they are legal instruments that define what manufacturers must demonstrate to be allowed to sell and operate vehicles in the relevant markets. Two regulations adopted by WP.29 in 2020 directly govern cybersecurity and software updates.

1.) **UNECE Regulation No. 155** establishes uniform provisions concerning the approval of vehicles regarding cybersecurity and the Cybersecurity Management System (CSMS). The regulation requires manufacturers to operate a certified CSMS covering the entire vehicle lifecycle, to perform risk assessments on each vehicle type, to implement appropriate mitigations against the threats listed in Annex 5, and to maintain monitoring and incident response capabilities throughout operation. Annex 5 of R155 is particularly relevant to software updates: it lists threat scenarios and required mitigations, including software signing, integrity verification, secure communications, and management of cryptographic keys. R155 explicitly recognises ISO/SAE 21434 as evidence of CSMS compliance.

2.) **UNECE Regulation No. 156** establishes the parallel provisions for software updates and the Software Update Management System (SUMS). It requires manufacturers to operate a certified SUMS covering the planning, deployment and recording of software updates, and defines specific requirements for the integrity, authenticity and traceability of each update, the identification of regulatory-relevant software (RXSWIN), and the documentation of update outcomes. R156 explicitly recognises ISO 24089 as evidence of SUMS compliance, making the relationship between regulation and technical standard explicit.

For UP2DATE4SDV, the practical effect of R155 and R156 is the following: the regulations make the CSMS and the SUMS mandatory for vehicle approval in the jurisdictions where they apply, and ISO/SAE 21434 and ISO 24089 are the recognised means of implementing and demonstrating them.

2.5. ISO 24089 - Software update engineering

ISO 24089, Road vehicles - Software update engineering, was published in February 2023 and is the central reference for the work described in this deliverable. It provides terminology, objectives, requirements and recommendations for software update engineering across the full vehicle lifecycle, from the organisational level down to individual software update campaigns. The standard is applicable to any organisation involved in software update engineering for road vehicles: vehicle manufacturers, suppliers, partners, and is intended as the common ground for managing activities and responsibilities across the supply chain.

ISO 24089 is structured around six technical clauses. Clauses 4 and 5 cover the organisational and project levels (governance, planning and supporting processes), and Clauses 8 and 9 cover the operational processes (the lifecycle of a software update package and of a software update campaign). The clauses most relevant to the work in this deliverable are **Clauses 6 and 7**, which define the functions the system must provide. These functions are organised into two zones:

- **Infrastructure (Clause 6):** the processes and information systems that manage software update operations outside the vehicle: cloud backends, repositories, package assembly tools, campaign orchestrators and the data stores that hold vehicle configuration information.
- **Vehicle and Vehicle Systems (Clause 7):** the functions that must exist within the vehicle to support update operations: receipt, integrity verification, condition checking, installation, activation, failure handling and the local management of safety and cybersecurity risks during the update.

Across both zones, ISO 24089 identifies four families of functions that must be present:

- **Managing risk:** identification, assessment and mitigation of safety and cybersecurity risks introduced by update operations.
- **Managing vehicle configuration information (VCI):** collection, storage, integrity maintenance and distribution.
- **Communicating software update campaign information:** notifications, results and user interaction.
- **Processing software update packages:** creation, verification, distribution, installation and activation.

ISO 24089 defines what functions must exist, but not where or how they should be implemented. Each organization can decide whether a given function runs in the cloud, in the Central ECU, in a gateway or in the affected edge device, and can place the same function at more than one level when needed. This flexibility is what allows UP2DATE4SDV to design its own architecture while staying conformant with the standard. The same four function families are also used to organize the SASE requirements in Chapter 3 and define a reference architectural in Chapter 4.

For UP2DATE4SDV, ISO 24089 is the structural reference of the entire WP4 work. Chapter 5. of this deliverable describes the update procedure as a sequence of phases that follows the process clauses of ISO 24089, extended with the SASE mechanisms [1] developed in the project. In other words, ISO 24089 defines what the update procedure for road vehicles should include, and UP2DATE4SDV defines how safety and security are enforced at each phase of this procedure.

3. SASE Requirements

The requirements that govern the UP2DATE4SDV software update procedure are derived from the normative landscape described in Chapter 2. Each requirement listed in Appendix A: Requirements: is traceable to one or more source standards and applies to one or more of the architecture levels introduced in Chapter 4. The requirements are organized according to the four families of functions that ISO 24089 identifies for software upgrade engineering. This grouping allows the requirements to remain aligned with the reference architecture and the upgrade procedure.

The complete set of requirements, with their full text, source standards is provided in Appendix A: Requirements.

3.1. Requirements Groups

Table 1 summarizes the four requirement groups, what each cover and the requirements assigned to it. Within each group, the full requirement text is given in Appendix A.

Table 1: SASE Requirements groups according to the ISO 24089 function families.

Group (ISO 24089 function)	Requirements
Managing risks	Conformance to SASE standards (REQ-SASE-001); Criticality-based partitioning (REQ-SASE-002); Pre-activation condition checks (REQ-SASE-004); Periodic system V&V (REQ-SASE-006); Supply-chain cybersecurity risk management (REQ-SASE-006-001); Continuous device monitoring (REQ-SASE-006-002); Vehicle-level cybersecurity validation (REQ-SASE-006-003); Startup self-diagnostics and fault reaction (REQ-SASE-006-004); In-vehicle resource validation (REQ-SASE-011); Simultaneous-access arbitration (REQ-SASE-012); Safety resilience under denial-of-service (REQ-SASE-012-001); Safe handling of failed updates (REQ-SASE-014); Hardware-backed key protection (REQ-SASE-015)
Managing vehicle configuration information (VCI)	VCI integrity and confidentiality (003); Protection against unauthorised modification (003-001)
Communicating software update campaign information	Secure update transmission (REQ-SASE-005); Authenticity and reliability of update delivery (REQ-SASE-005-001); Communication logging and intrusion monitoring (REQ-SASE-005-002); Tamper-evident update logging (REQ-SASE-013-003)
Processing software update packages	Software authenticity verification (REQ-SASE-007); Authenticated boot (REQ-SASE-008); Package compatibility validation (REQ-SASE-009, REQ-SASE-009-001, REQ-SASE-009-002); Dependency validation and regression testing (REQ-SASE-010); Update V&V (REQ-SASE-013); Post-update multi-level safety validation (REQ-SASE-013-001); Post-update self-diagnostics (REQ-SASE-013-002); Pre-release verification and validation (REQ-SASE-016)

4. UP2DATE4SDV reference architecture

4.1. Purpose and scope

This chapter presents the UP2DATE4SDV reference architecture based on the two-zone model defined by the ISO 24089 standard: Infrastructure (Clause 6) and Vehicle and vehicle system (Clause 7). It also shows the structure of the components and their distribution within the system. Section 4.2 introduces the UP2DATE4SDV reference architecture: a centralised zonal E/E architecture organised in four levels, with the logical components hosted at each level. Section 4.2.2 explicitly maps the standard's two zones and four function families onto the four architectural levels.

4.2. Proposed Architecture UP2DATE4SDV

4.2.1. Architectural rationale: a centralised zonal E/E architecture

Vehicle E/E architectures have changed significantly over the last decades [15]. In early vehicles, each function had its own dedicated ECU. As the number of electronic functions increased, this distributed architecture resulted in a high number of ECUs and complex wiring. The industry then moved to a domain-based architecture, where ECUs are grouped by functional domain (powertrain, chassis, body, infotainment, ADAS) under a domain controller. The most recent step, driven by the software-defined vehicle, is the centralised zonal architecture. Here, functions are grouped by their physical location in the vehicle instead of by domain, and computing power is concentrated in a few high-performance units. This approach reduces wiring, makes it easier to integrate new functions, and provides the centralised computing and well-defined update points needed for software-defined vehicles.

UP2DATE4SDV adopts a centralised zonal electrical and electronic (E/E) architecture (see Figure 4) for the in-vehicle system, aligned with the dominant trend of software-defined vehicles. In this architecture the following components will be found:

- A high-performance Central ECU: concentrates the compute, the application logic and the decision-making that used to be spread across several domain ECUs;
- A set of Zonal Gateways: group the physical interfaces, sensors, actuators and edge ECUs by physical location in the vehicle rather than by functional domain, each gateway acting as a regional aggregator between the Central ECU and its zone;
- Edge Devices: are limited to time-critical and hardware-specific functions, while higher-level processing and orchestration are delegated to the zonal gateway and the central ECU.

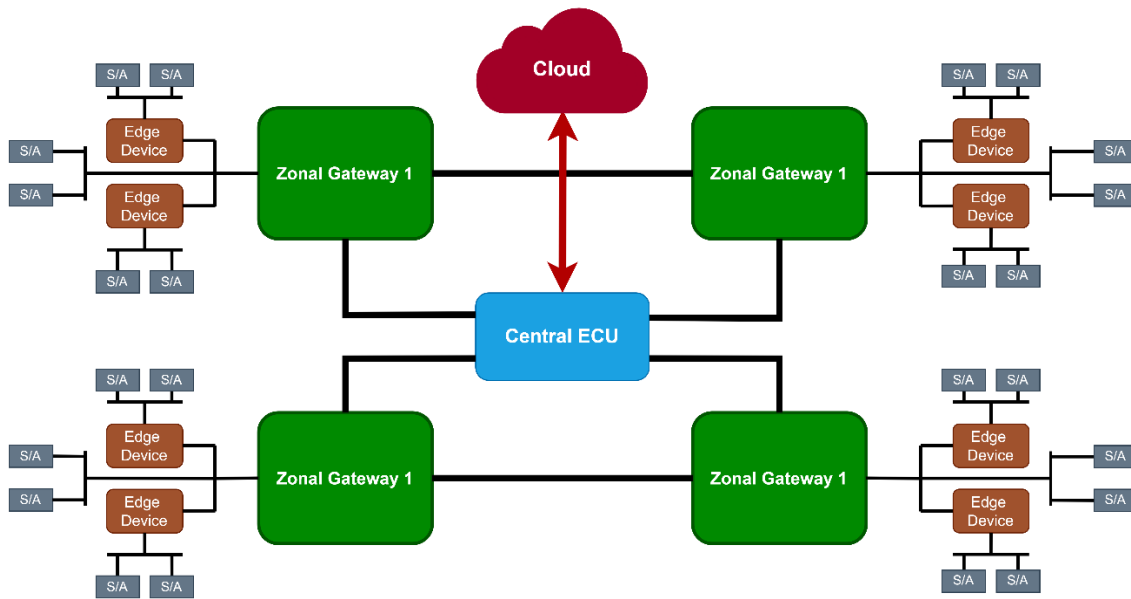


Figure 4: Centralised zonal E/E Architecture.

This architectural choice has direct consequences for software update engineering. First, it concentrates the orchestration of updates in the Central ECU and the zonal gateways, instead of distributing it across many heterogeneous ECUs. This simplifies the management of vehicle configuration information, the verification of dependencies and the enforcement of safe vehicle state. Second, it allows SASE guarantees to be applied consistently at every architectural level. Third, it scales incrementally: adding a new edge device requires updating the corresponding zonal gateway but not the central platform or the cloud. The architecture extends the in-vehicle structure with a Cloud level that hosts the infrastructure functions of ISO 24089. Together, the cloud and the in-vehicle structure form a single end-to-end architecture organised in four levels presented in the next section.

4.2.2. Architecture overview

The reference architecture is structured into four levels, of which any of levels 2, 3 and 4 can be updated, as shown in Figure 5:

- **Level 1 - Cloud:** Hosts the infrastructure functions of the system. It maintains the master vehicle configuration information, orchestrates software update packages and campaigns, and supervises safety and security (SASE) at the infrastructure level.
- **Level 2 - Central ECU:** Represents the central computing platform of the vehicle. It receives update packages from the cloud, distributes them within the vehicle, maintains the vehicle-level configuration information, supervises in-vehicle SASE, and manages interaction with the vehicle user.
- **Level 3 - Zonal Gateway:** Acts as an intermediate aggregation and distribution layer. It forwards relevant parts of the update package to the edge devices within its zone, maintains zone-level configuration information, and supervises SASE at the zonal level. A vehicle may contain several zonal gateways, shown in Figure 5 as ZG1, ZG2, ZG3, and ZG4.
- **Level 4 - Edge Device / ECU:** Hosts the software components closest to sensors and actuators. It represents the last level of the architecture, and depending on whether it is being updated, each device receives, installs, and activates the update in its own software instance, implementing a reduced set of orchestration, configuration, and SASE functions.

The architecture is composed of a set of logical components, instantiated at each level with different responsibilities: the SUP Orchestrator, the SUP Executor, the VCI Registry and the SASE Monitor, together with two level-specific components, the Campaign Orchestrator (only at L1) and the User Interaction Handler (only at L2). Figure 5 shows their distribution across the four levels, and the remaining subsections describe the role of these components level by level.

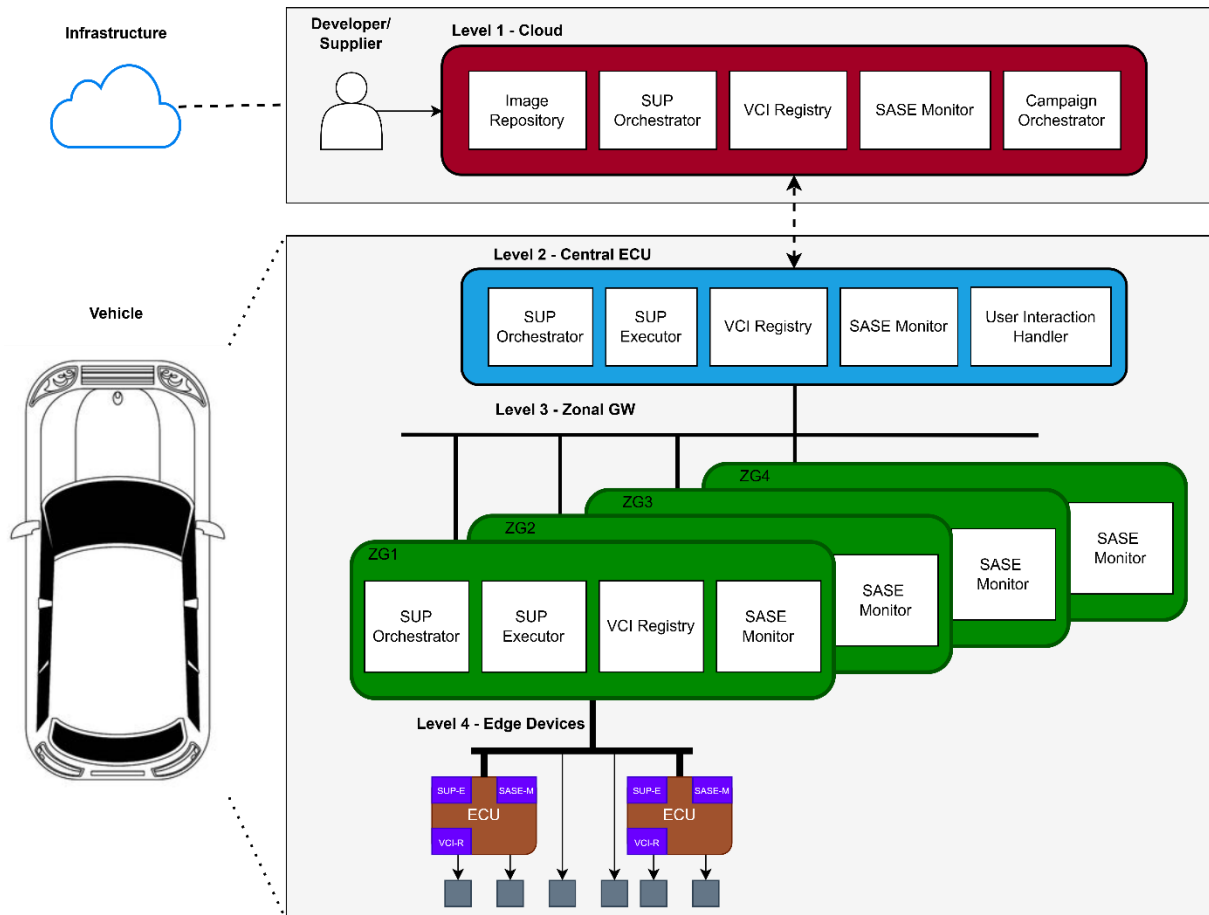


Figure 5: Proposed Architecture UP2DATE4SDV.

4.3. Architecture Levels

4.3.1. Level 1 – Cloud

The cloud level implements the infrastructure side of ISO 24089 (Clause 6). It hosts the following logical components, summarized in Figure 6:

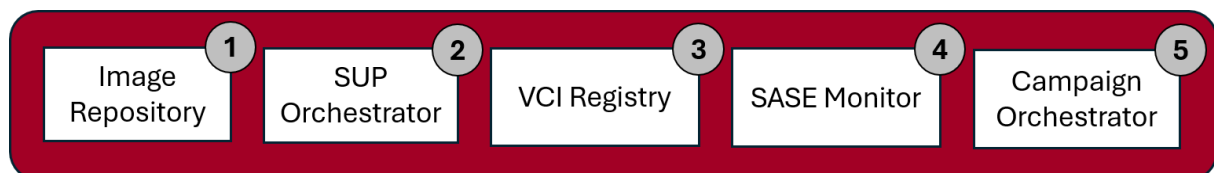


Figure 6. Cloud Sub-Components.

- 1.) **Image Repository:** Stores the software images and update packages available for distribution, keeping them versioned and integrity-protected until they are selected for a campaign.

- 2.) **SUP Orchestrator:** Creates, signs and distributes software update packages. Implements the package lifecycle, selection of contents, assembly, verification and validation, and approval for release, and the distribution function.
- 3.) **VCI Registry:** Stores the configuration information of every vehicle in the fleet. Holds the master record that the SUP Orchestrator uses to identify which vehicles and devices must receive each update, to check compatibility and to identify dependencies.
- 4.) **SASE Monitor:** Supervises the safety and cybersecurity of the cloud infrastructure itself. Manages cybersecurity risks of the infrastructure, enforces the cybersecurity actions identified for each package, and oversees the integrity and authenticity of the artefacts handled at the cloud level.
- 5.) **Campaign Orchestrator:** Plans and coordinates software update campaigns end-to-end, implementing the preparation, execution and completion activities. Selects approved packages, determines targets and distribution methods, defines corrective actions, drives notifications to the related parties and collects results.

4.3.2. Level 2 – ECU Central Layer

The Central ECU is the entry point of the update flow into the vehicle and the level that holds the in-vehicle authority for software update operations. It hosts:

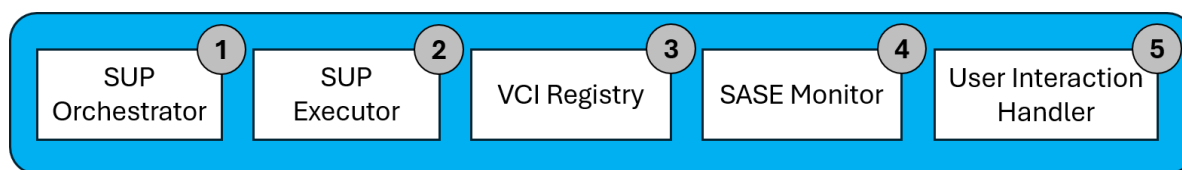


Figure 7. ECU Central Sub-components.

- 1.) **SUP Orchestrator:** Receives the update package from the Cloud and forwards the relevant parts to the affected Zonal Gateways in case the update does not affect the Central ECU itself. Acts as the in-vehicle counterpart of the cloud SUP Orchestrator: it preserves integrity and authenticity end-to-end, arbitrates simultaneous access requests, and coordinates the receipt, installation and activation steps across the vehicle. \
- 2.) **SUP Executor:** Receives the update package on the Central ECU's own software and installs and activates it, including signature verification just before activation, and local rollback if it fails.
- 3.) **VCI Registry:** Holds the consolidated vehicle-level record of the configuration for each ECU and each zone of the vehicle, a complete copy on the vehicle, not the authoritative source for the entire fleet, which remains in the cloud. This record is exposed to the Central SUP Orchestrator to perform compatibility and dependency checks during upgrade operations. After a successful campaign, it consolidates the new status reported by the zonal gateways and edge devices and synchronises the resulting record with the Cloud VCI Registry.
- 4.) **SASE Monitor:** Supervises safety and cybersecurity at the vehicle level, both continuously and during update operations. It enforces the safe vehicle state required for each step of an update, coordinates with relevant safety and security systems to maintain or restore a safe state, and monitors the cybersecurity properties of the package as it moves through the vehicle.
- 5.) **User Interaction Handler:** Handles user interaction when the update process requires it. ISO 24089 mentions user interaction as part of the process but does not make it mandatory. When used, it

notifies the user (the driver) of the availability, criticality and implications of the campaign, obtains the confirmation needed before activation, and informs the user about the outcome.

4.3.3. Level 3 – Zonal Gateway Layer

Each Zonal Gateway hosts four logical components, instantiated per zone. These components implement the same ISO 24089 functions described in Levels 1 and 2, but applied at the zonal level:

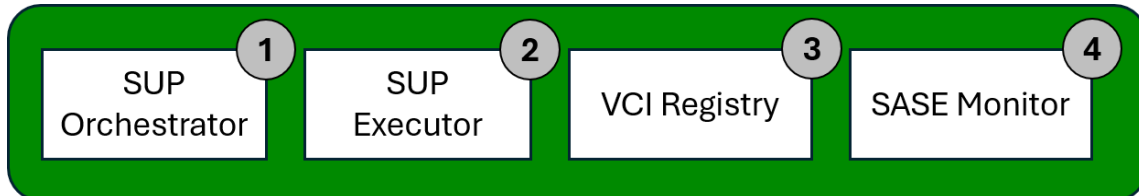


Figure 8. Zonal Gateway Sub-Components.

- 1.) **SUP Orchestrator:** Receives the relevant update package from the Central ECU and distributes it to the edge devices in its zone, coordinating the receipt, installation and activation within the zone, in case the update does not affect the Zonal Gateway itself.
- 2.) **SUP Executor:** Receives the update package on the Zonal Gateway's own software and installs and activates it, including signature verification just before activation, and local rollback if it fails.
- 3.) **VCI Registry:** Holds the zone-level subset of vehicle configuration information: the inventory of edge devices in the zone and their current software versions. Synchronises with the ECU Central VCI Registry.
- 4.) **SASE Monitor:** Supervises safety and cybersecurity at the zone level. Detects abnormal behaviour of edge devices during update operations and either applies corrective actions locally or escalates them to the Central SASE Monitor.

Because there is more than one Zonal Gateway per vehicle, Figure 5 shows ZG1, ZG2, ZG3 and ZG4, the four components are replicated per zone, and the SUP Orchestrator of the Central ECU coordinates them when an update covers multiple zones.

4.3.4. Level 4 – Edge Devices Layer

The Edge level is where the update is finally written and activated. Each edge device hosts three logical components in Figure 9.

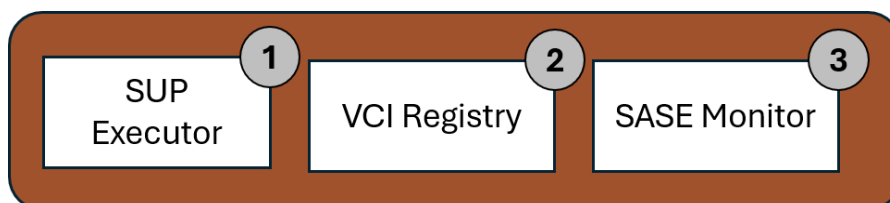


Figure 9. Edge Devices Sub-components.

- 1.) **SUP Executor:** Receives the update package on the device and installs and activates it, including signature verification just before activation, and local rollback if it fails.
- 2.) **VCI Registry:** Holds the configuration information of the device itself (software versions, hardware identifiers, configuration parameters) and reports them up the chain.

- 3.) **SASE Monitor:** Ensures security and cybersecurity at the device level maintains safe operation during installation and activation, enables rollback in case of failure, and reports abnormal events to the SASE Monitor of the zone.

4.4. Mapping ISO 24089 with UP2DATE4SDV Architecture

The mapping between ISO 24089 and the UP2DATE4SDV architecture follows directly from the previous section. The standard's infrastructure zone (Clause 6) is implemented by Level 1: Cloud and the standard's vehicle and vehicle systems zone (Clause 7) is realised jointly by Levels 2 to 4: Central ECU, Zonal Gateway and Edge Device.

Table 2: ISO 24089 to UP2DATE4SDV Architecture Mapping

Standard Section (Clauses 6/7)	L1 - Cloud	L2 – Central ECU	L3 – Zonal Gateway	L4 – Edge Devices
Managing risk (6.3.1 / 7.3.1)	Infrastructure cybersecurity risks (SASE Monitor)	Vehicle-level safety and cybersecurity: safe vehicle state (SASE Monitor)	Zone-level safety and cybersecurity (SASE Monitor)	Device-level safety and cybersecurity: fallback / rollback (SASE-M)
Managing VCI (6.3.2 / 7.3.2)	Fleet-level VCI record (VCI Registry)	Vehicle-level VCI record (VCI Registry)	Zone-level VCI record (VCI Registry)	Device-level VCI record (VCI Registry)
Communicating campaign information (6.3.3 / 7.3.3)	Campaign planning; related-party communications (Campaign Orchestrator)	Vehicle-user interaction; confirmation and outcome reporting (User Interaction Handler)	-	-
Processing SW Update packages (6.3.4 / 7.3.4)	Creation, signing, distribution; target resolution (SUP Orchestrator)	In-vehicle reception, forwarding, arbitration; integrity preservation (SUP Orchestrator)	Zone-level distribution and coordination (SUP Orchestrator)	Local receipt, installation, activation; signature & compatibility verification (SUP-O)

4.5. UP2DATE4SDV Architecture Implementation in WP3

As explained in Section 1.1, WP4 is supporting the other technical work packages. In D3.1[1] the proposed architecture for the Orchestration and Reconfiguration Plane (ORP) was presented, based on the collected requirements from WP5 and the refinement in WP3. This technical view on the architecture in Figure 10 shows components for Update Development, Cloud and the Automotive Platform. Update Development, which is not covered in this document, is detailed in D3.2[2] of WP3.

The relevant deployment part of WP3 provides a clear mapping to the architecture presented in this deliverable.

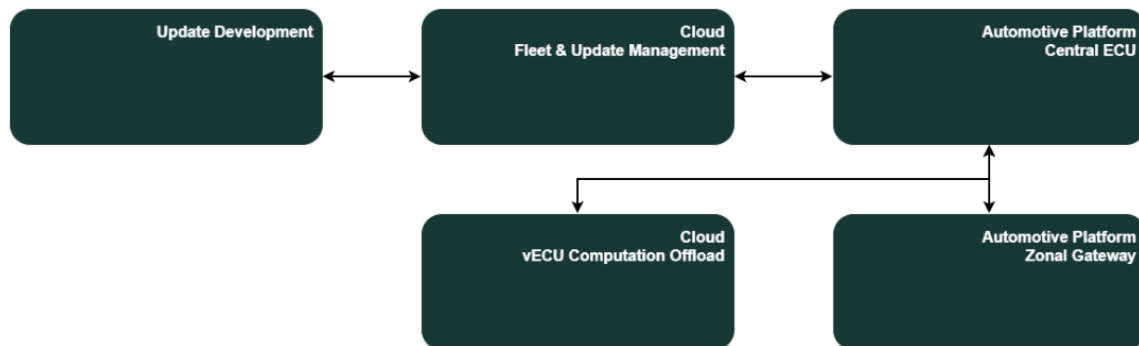


Figure 10: Architecture of the ORP[1]

In WP3, deliverable D3.1[1] is followed by D3.3[3], which provides a more technical refinement of the architecture, mapping it to implementations from project partners.

5. Update procedure

Building on the reference framework introduced in Chapter 4, this chapter describes how UP2DATE4SDV instantiates the software update procedure with safety and security (SASE) considerations integrated end-to-end. The procedure is presented along the two physical zones identified in the framework: the infrastructure

5.1. Infrastructure Process

5.1.1. Purpose

The infrastructure side of the update procedure covers the full lifecycle of a software update from the moment new software arrives from the software provider until the moment a campaign is formally closed and its outcomes feed back into the organisation's continuous improvement cycle. From a SASE standpoint, the infrastructure is where most of the upstream guarantees are established: package integrity and authenticity are built here, cybersecurity risks of the OTA backend are managed here, and the decisions that protect the vehicle from unsafe or unauthorised updates are taken here before any byte travels to the vehicle.

The procedure described in this subchapter follows the ISO 24089 process clauses (Clauses 4, 5, 8 and 9) on the infrastructure side, invoking the capabilities of Clause 6 wherever the process needs them.

5.1.2. Process overview

The infrastructure process is structured into six sequential phases, each producing one or more work products that act as the contract for the next phase. The phases are derived from a temporal grouping of ISO 24089 Clauses 4, 5, 8 and 9, and they invoke the four function families of Clause 6 (managing risk, managing vehicle configuration information, communicating campaign information, processing software update packages) wherever needed.

The six phases are:

- **Phase 0 - Preliminary framework:** Organisational and project-level processes, plus continuous management of infrastructure cybersecurity risks. This phase enables every subsequent activity and applies transversally throughout the lifecycle.
- **Phase 1 - Package ingestion and preparation:** Reception of software from the supply chain, continuous management of vehicle configuration information, identification of targets and contents, package assembly, verification and validation, and formal release approval. As result of this phase the *approved software update package* is produced.
- **Phase 2 - Campaign preparation:** Definition of campaign purpose, scope, conditions, dependencies, risks and user interaction, consolidated into a *software update campaign plan*.
- **Phase 3 – Distribution:** Resolution of targets into individual recipients, pre-distribution verification gate, package delivery preserving integrity and authenticity, and pre-activation communication with the vehicle user.
- **Phase 4 - Monitoring and failure management:** Reception of execution status and results from recipients, failure detection and corrective actions, and post-operation communication to related parties.
- **Phase 5 - Closure:** Campaign execution records, closure communication, and feedback into continuous improvement.

The flow between phases is shown in Figure 11, where each transition is labelled with the work product handed over from one phase to the next.

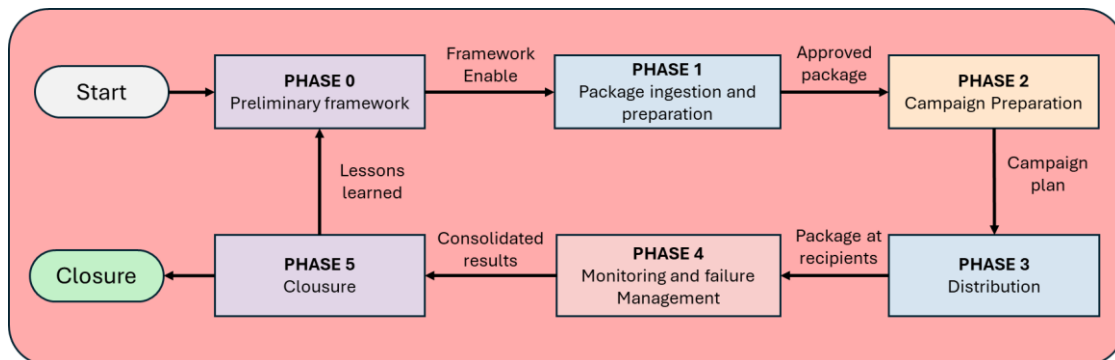


Figure 11. End-to-end Infrastructure update process.

5.1.3. Safety and Security

ISO 24089 defines what each phase of the update process must do, but it does not prescribe how safety and security shall be enforced across that process. UP2DATE4SDV addresses that gap by introducing a transversal SASE layer, materialised as a logical SASE Monitor that supervises every phase of the infrastructure process. The SASE Monitor is not an additional phase but a continuous capability that intersects all of them: at each point of the lifecycle, it identifies what assets are exposed, which threats apply, and which safety and cybersecurity guarantees must be enforced. Figure 12 illustrates this transversal supervision over the six phases.

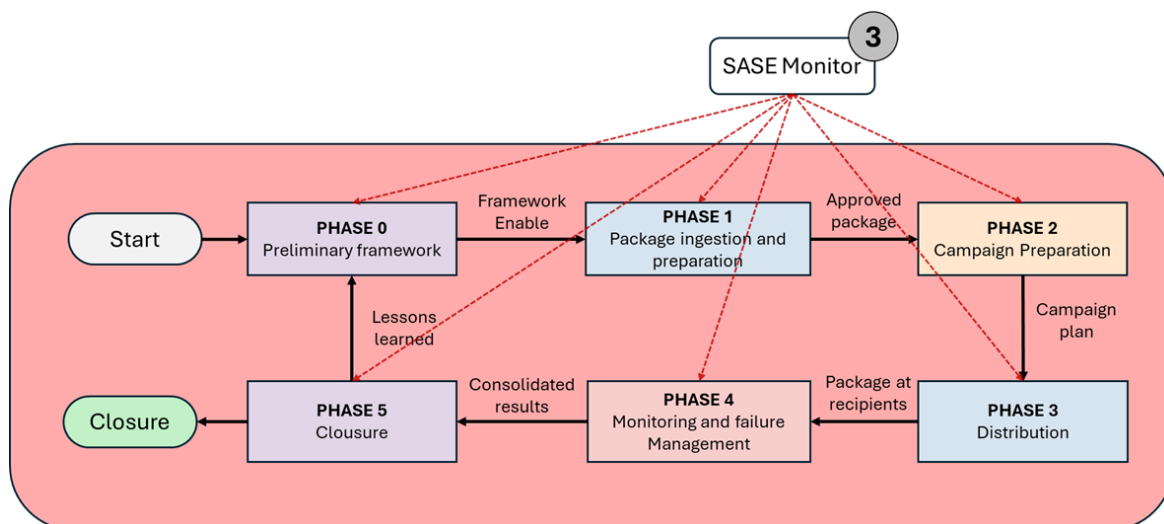


Figure 12. SASE Monitor as a transversal supervision layer over the six phases of the infrastructure update process.

The protection strategy is built on three normative pillars: ISO 26262 for functional safety, ISO/SAE 21434 for cybersecurity engineering, and UNECE R155/R156 for regulatory compliance, complemented by IEC 62443 where industrial communication and zoning are involved.

These standards are the basis for the SASE requirements of the project. Table 3 maps each phase of the infrastructure process to the requirements enforced within it; the full text of each requirement is given in Appendix A: Requirements..

Table 3: Mapping between the infrastructure update phases and the SASE requirements.

Phase	Total	Requirements ID
P0 - Preliminary framework	2	REQ-SASE-001, REQ-SASE-006-001
P1 - Package ingestion and preparation	8	REQ-SASE-003, REQ-SASE-003-001 REQ-SASE-006, REQ-SASE-009-001 REQ-SASE-010, REQ-SASE-013 REQ-SASE-015, REQ-SASE-016
P2 - Campaign preparation	3	REQ-SASE-004, REQ-SASE-009-001 REQ-SASE-010
P3 - Distribution	5	REQ-SASE-003, REQ-SASE-005 REQ-SASE-005-001, REQ-SASE-005-002 REQ-SASE-012-001
P4 - Monitoring and failure management	2	REQ-SASE-005-002, REQ-SASE-013-003
P5 - Closure	1	REQ-SASE-013-003

5.2. Vehicle Process

5.2.1. Purpose

The vehicle side of the update procedure covers everything that happens within the vehicle from the moment it begins receiving a software update package until it has reported the outcome of the update operation back to the infrastructure. From a SASE standpoint, the vehicle is where the upstream guarantees built in the infrastructure must be enforced in real conditions: integrity and authenticity verifications must execute on the recipient itself, conditions for safe operation must be confirmed against the live state of the vehicle, the activation must preserve safety throughout, and any failure must be handled in a way that keeps the vehicle in a safe state.

The vehicle process runs across the three in-vehicle levels defined in the architecture: the Central ECU (L2), the Zonal Gateways (L3) and the Edge Devices (L4). The update package enters the vehicle through the Central ECU (L2), which acts as the entry point from the infrastructure and forwards the package towards its target. The Zonal Gateways (L3) distribute the package within their zone and coordinate the devices connected to them. The Edge Devices (L4) are the final targets where the update is installed and activated. The target of a campaign may be at any of these levels; each level runs the phases of the process that correspond to the part it plays in each update.

5.2.2. Node roles: Bridge and recipient

The role that each node plays depends on the target of the campaign. Two roles are distinguished: the bridge role and the recipient role. The role is decided per campaign, not fixed per node: the same Central ECU acts as recipient for an update targeted at it, and as bridge for an update targeted at a Zonal Gateway or an Edge Device below it.

A.) **Bridge role:** A node acts as a bridge when the package passes through it on its way to a node at a lower level. A bridge receives the package, verifies its integrity and authenticity, and forwards it to the next node along the path. It does not install or activate the update. Because the package is verified at every bridge, a tampered package is stopped as early as possible on the path, before it reaches the

recipient. In this role the node runs only the phases related to reception, storage and verification, handled by its SUP Orchestrator.

B.) **Recipient role:** A node acts as a recipient when it is the destination of the campaign. A recipient performs the same reception and verification steps as a bridge and, in addition, runs the phases related to installation, activation and post-activation monitoring, handled by its SUP Executor. The recipient therefore runs the full vehicle-side update lifecycle.

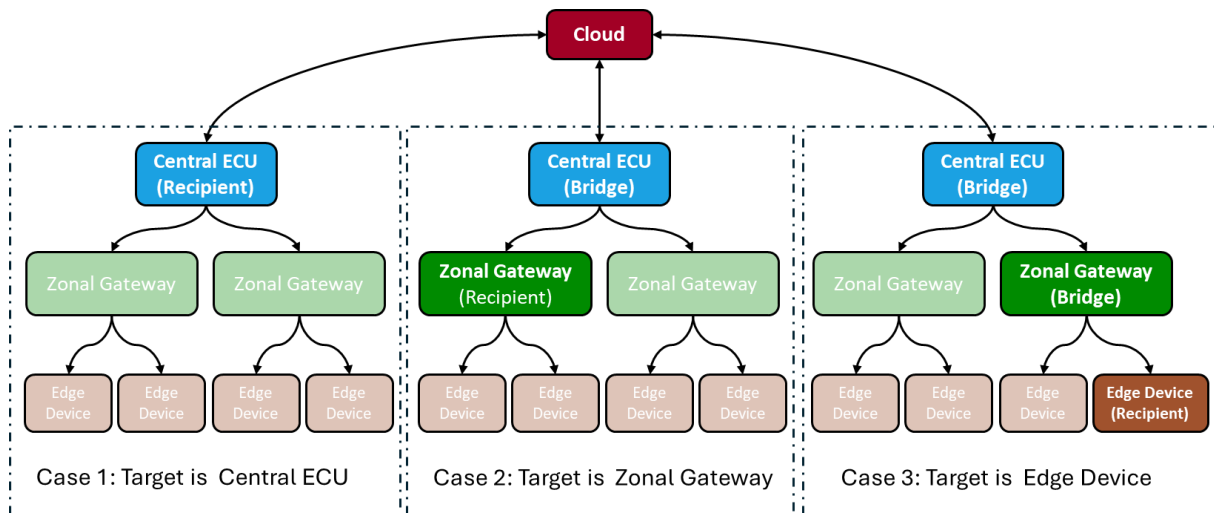


Figure 13. Bridge and recipient roles depending on the campaign target.

In each case (See Figure 13), the campaign target acts as recipient and runs the full update lifecycle; the nodes on the path between the cloud and the recipient act as bridges; the remaining nodes do not participate in the campaign.

5.2.3. Process overview

The vehicle process is structured into seven sequential phases. Phase V0 is a transversal framework phase, equivalent in role to Phase 0 of the infrastructure process, and the remaining six phases correspond to the operational sequence experienced by an individual vehicle as it goes through a software update.

The seven phases are:

- **Phase V0 - Vehicle readiness framework:** Continuous management of safety and cybersecurity risks within the vehicle, continuous maintenance of vehicle configuration information, and continuous availability of communication functions with infrastructure and user. This phase enables every operational phase and applies transversally throughout the lifecycle.
- **Phase V1 - Reception:** Reception of the software update package through one of the supported distribution methods, handling of communication interruptions during reception, and storage of the received package in a state where it is not yet installed.
- **Phase V2 - Pre-activation verification:** Verification of integrity and authenticity of the received package, confirmation of compatibility with the current vehicle configuration, evaluation of the necessary conditions for the update operation including safe vehicle state and resource availability, and arbitration of simultaneous access requests.

- **Phase V3 – Installation:** Writing of the relevant parts of the package to the target ECU storage while preserving safe vehicle state, leaving the package installed but not yet executable.
- **Phase V4 - Activation.** Transition of the installed package to executable state. The temporal limits imposed by the standard apply here: the verifications of conditions, integrity and authenticity, and compatibility must have completed at the latest prior to this point.
- **Phase V5 - Post-activation monitoring and failure handling.** Monitoring of vehicle behaviour after activation, detection of failures or unsafe operation, and execution of corrective actions including rollback to the previous version, transition to a safe or degraded state, or restriction of vehicle functionality, to ensure vehicle safety in all circumstances.
- **Phase V6 - Reporting and information.** Communication to the vehicle user of the outcome and any associated information, reporting of execution status and results to the infrastructure, and confirmation of cycle closure on the vehicle side.

The flow between phases is shown in Figure 14, where each transition is labelled with the operational state handed over from one phase to the next.

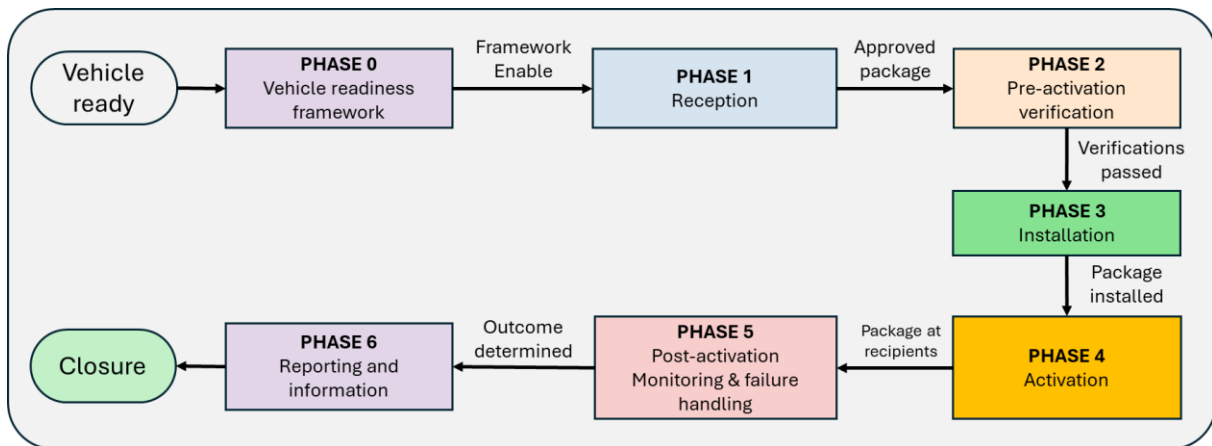


Figure 14. Vehicle Process based on ISO 24089.

5.2.4. Safety and Security

ISO 24089 defines what each phase of the vehicle update process must do, but it does not prescribe how safety and security shall be enforced on the recipient side. UP2DATE4SDV addresses that gap by extending the transversal SASE layer from the infrastructure side to the vehicle: a logical SASE Monitor supervises every phase from V0 to V6, identifying at each point what assets are exposed, which threats apply, and which guarantees must be enforced within the vehicles. Figure 15 illustrates this transversal supervision.

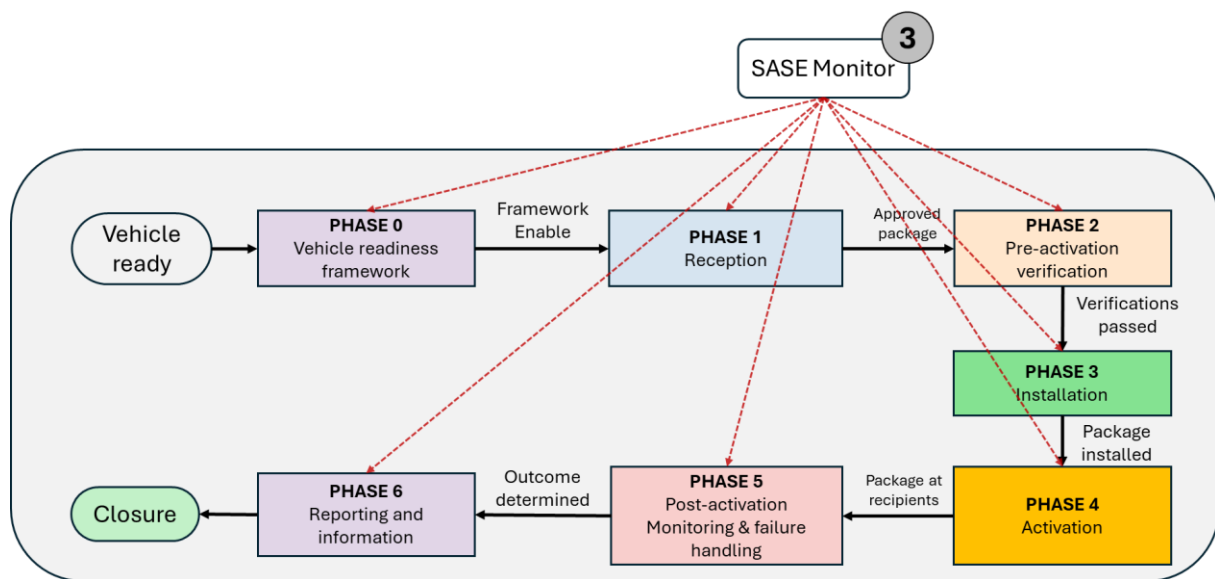


Figure 15. SASE Monitor over the vehicle update process.

The protection strategy on the vehicle side rests on the same three normative pillars as the infrastructure side, ISO 26262 for functional safety, ISO/SAE 21434 for cybersecurity, UNECE R155/R156 for regulatory compliance, but the emphasis shifts: where the infrastructure protected the package as it was being built and dispatched, the vehicle must protect the package as it is verified and executed in real conditions, and protect the vehicle itself during and after the operation

These standards are the basis for the SASE requirements enforced on the vehicle side. Table 4 maps each phase of the vehicle process to the requirements enforced within it; the full text of each requirement is given in Annex A: Requirements.

Table 4: Mapping between the vehicle update phases and the SASE requirements.

Phase	Total	Requirements ID
V0 - Vehicle readiness framework	4	REQ-SASE-002, REQ-SASE-006-002, REQ-SASE-006-004
V1 - Reception	1	REQ-SASE-005, REQ-SASE-005-001
V2 - pre-activation verification	5	REQ-SASE-004, REQ-SASE-007, REQ-SASE-009, REQ-SASE-010, REQ-SASE-009-002, REQ-SASE-011
V3 - Installation	2	REQ-SASE-003-001, REQ-SASE-012, REQ-SASE-012-001, REQ-SASE-015
V4 - Activation	2	REQ-SASE-004, REQ-SASE-008
V5 - post-activation monitoring and failure handling	5	REQ-SASE-006-002, REQ-SASE-006-003, REQ-SASE-013 REQ-SASE-013-001, REQ-SASE-013-002 REQ-SASE-014
V6 - Reporting and Information	1	REQ-SASE-013-003

5.3. UP2DATE4SDV Process Implementation in WP3

As with the architecture in Section 4.5, the process is also closely connected to the work in WP3. In D3.1[1] a first definition of the update process in the ORP was provided, focused on a detailed flow from development of a change during development up to the deployment and validation inside the vehicle. This process is covered by the more abstract processes defined in Sections 5.1 (Infrastructure) and 5.2 (Vehicle). In contrast, D3.2[2] and D3.3[3] deliver a more technical refinement of the same process from D3.1. While further progressing with the development of the OPR, WP3 will integrate the requirements defined by WP4 and adjust to the outlined processes. This will lead to a further refinement of the original process from D3.1, which is expected to be provided in a final version in D4.4 in M30 of the project.

Appendix A: Requirements:

ID	Requirement	Source	Architecture
REQ-SASE-001	The software update system shall conform to ISO 24089, ISO 26262 (parts -6 and -8), ISO/SAE 21434 and IEC 62443-4 in the parts applicable to OTA infrastructure and in-vehicle update mechanisms.	ISO 26262-6; ISO 26262-8; IEC 62443; ISO/SAE 21434	L1, L2, L3, L4
REQ-SASE-002	In the vehicle data, applications and services shall be partitioned based on their criticality.	ISO 26262-6 Clause 7; ISO 26262-9	L2, L3, L4
REQ-SASE-003	The system shall provide one or more functions to maintain integrity (at rest and in runtime) and confidentiality (at rest) of collected vehicle configuration information as well as the software update and metadata.	ISO 24089	L1, L2, L3, L4
REQ-SASE-003-001	The system shall prevent unauthorised or unintended modification of vehicle related information (software, configuration, metadata). At rest or at runtime.	UNECE R155 UNECE R156	L1, L2, L3, L4
REQ-SASE-004	The system shall provide one or more functions to verify that all organisation-defined conditions required for a software update campaign are met prior to activation of the software update. This includes verifying for example that power is sufficient, ensuring that interrupting the software under update is safe and assessing that runtime updates do not impact system safety.	ISO 24089 ISO 26262-3	L2, L3, L4
REQ-SASE-005	Update shall be transmitted via secure communication based on IEC 62443.	IEC 62443	L1, L2, L3, L4
REQ-SASE-005-001	The system shall ensure secure, authorised, and reliable software updates by preserving the authenticity, confidentiality, and integrity of update packages, supporting safe communication mechanisms, handling communication interruptions, and preventing any unauthorised or irrelevant message transmission.	ISO 24089 IEC 61508	L1, L2, L3, L4
REQ-SASE-005-002	The system shall provide communication logging and real-time monitoring of inter-node communications to support intrusion detection, including packet capturing capabilities, anomaly detection, and protection of log storage from external manipulation.	IEC 62443 ISO/SAE 21434	L1, L2, L3
REQ-SASE-006	The functionality and appropriateness of the system shall be periodically verified and validated.	ISO 24089	L1, L2, L3, L4
REQ-SASE-006-001	The cybersecurity risks of the OTA backend and vehicle infrastructure and the full supply chain (including components and third-party software), including threat assessment and vulnerability handling shall be managed periodically.	ISO/SAE 21434 Cl.8, Cl.13; ISO/IEC 27000;	L1, L2, L3, L4

		UNECE R155 Cl.7.2.2.3, Cl.7.3.2	
REQ-SASE-006-002	The system shall support continuous monitoring of end-devices by providing access to relevant operational properties and performance metrics, including observation points, traces, performance counters, processor-utilization data, built in tests, and power-consumption estimations, so that running softwares can be effectively observed and analyzed (such as CPU, memory, communication bandwidth, and partition or HSM availability)	ISO 24089; ISO 26262	L1, L2, L3, L4
REQ-SASE-006-003	Cybersecurity validation shall be performed at vehicle level to confirm that cybersecurity goals are achieved, and cybersecurity claims are supported.	ISO/SAE 21434	L2, L3, L4
REQ-SASE-006-004	The system shall implement built-in self-diagnostic and self-monitoring mechanisms to detect random hardware faults and relevant systematic faults at startup and shall initiate defined fault-reaction measures upon detection of a failure.	ISO 26262-11	L2, L3, L4
REQ-SASE-007	The system shall ensure that only authentic, integrity-protected, and non-malicious software is loaded or executed by verifying software update packages before activation.	IEC 62443-4-2 EDR 3.10	L1, L2, L4
REQ-SASE-008	Authenticity of the boot process: the system shall verify the authenticity of the firmware, software, and configuration data needed for the boot process prior to it being executed.	IEC 62443-4-2 EDR 3.14	L2, L3, L4
REQ-SASE-009	The system shall assess and validate that each software update package is compatible with the target system's software and hardware configuration, ensuring that compatibility is identified and confirmed.	ISO 24089 UNECE R156	L1, L2, L4
REQ-SASE-009-001	The cloud shall assess and validate that each software update package is compatible with the target system's software and hardware configuration, ensuring that compatibility is identified and confirmed prior to deployment.	ISO 24089 UNECE R156	L1
REQ-SASE-009-002	The vehicle shall assess and validate that each software update package is compatible with the target system's software and hardware configuration, ensuring that compatibility is identified and confirmed prior to activation.	ISO 24089 UNECE R156	L2, L4
REQ-SASE-010	The system shall identify and validate all dependencies and interdependencies associated with a software update package, ensuring that interactions with other system components are correctly managed and that integration testing confirms the update introduces no regressions prior to deployment.	ISO 24089	L1, L2, L4

REQ-SASE-011	The system shall identify and validate the in-vehicle resources required by the target for software update activities prior to deployment, ensuring that resource usage (such as CPU, memory, communication bandwidth, and partition or HSM availability) does not impair real-time safety functions, and shall provide access to performance monitoring mechanisms to support effective oversight of these resources.	ISO 24089 ISO 26262	L1, L2, L4
REQ-SASE-012	Simultaneous access requests to vehicle during the software update operation shall be managed to maintain safe vehicle state.	ISO 24089	L2, L3, L4
REQ-SASE-012-001	The system shall be able to maintain essential safety services and functions during a denial-of-service event, ensuring that update-related traffic or attacks cannot starve safety-critical real-time tasks of required computational, communication, or memory resources.	IEC 62443-4-2 CR 7.1	L2, L3, L4
REQ-SASE-013	The functionality and appropriateness of the software update shall be verified and validated.	UNECE 7.1.3.3 R155 and R156	L1, L2, L3, L4
REQ-SASE-013-001	The system shall ensure that integration-, system-, and vehicle-level validation is performed after software updates, providing evidence that safety requirements, timing and performance constraints, technical safety requirements, robustness, and correct interactions between components are preserved, and that vehicle-level safety goals and mitigation strategies remain satisfied post-update. To do so the system shall provide access to relevant operational properties and performance metrics including observation points, traces, built in tests, performance counters, processor-utilization data, and power-consumption estimations.	ISO 26262	L2, L3, L4
REQ-SASE-013-002	The system shall implement built-in self-diagnostic and self-monitoring mechanisms to detect random hardware faults and relevant systematic faults as post-update verification tests, and shall initiate defined fault-reaction measures upon detection of a failure.	ISO 26262	L2, L3, L4
REQ-SASE-013-003	The system shall record every software update event with tamper-evident logging, obtain the execution status for each update recipient, and provide clear information regarding the success or failure of the update as well as any changes introduced. Campaign completion records, including purpose, targets, updated contents, start date, end date, and results, shall be managed and stored.		L1, L2
REQ-SASE-014	The system shall ensure vehicle safety in the event of a failed or interrupted software update by executing predefined corrective actions, such as rollback to previous version, transition to a safe or known secure state, or activation of safety mechanisms, to guarantee	ISO 24089 Clause 7.3.4.10	L2, L3, L4

	proper system behaviour and maintain functional safety during update or reconfiguration malfunctions.		
REQ-SASE-015	The system shall provide a hardware-backed protected environment to ensure the confidentiality, integrity, and authenticity of cryptographic keys and sensitive data used for software update operations, including signature verification, decryption, and secure key storage.	IEC 62443 EDR 3.12	L2, L3, L4
REQ-SASE-016	Before the release of the software update package the system shall run complete verification and validation to ensure it contains only the intended software and metadata, and it shall not be released until successful verification and validation confirm its correctness and authorise its approval for deployment.	ISO 24089	L1