



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

D1.1. Node-level SASE-relevant properties V1.0

BARCELONA SUPERCOMPUTING CENTER (BSC)

31.12.2025



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. 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	D1.1	
Deliverable Name	Node-level SASE-relevant properties	
Nature	Report (R)	
Dissemination Level	Public	
Contractual Date of Delivery	31.12.2025	
Actual Submission Date	31.12.2025	
Lead Beneficiary	BARCELONA SUPERCOMPUTING CENTER (BSC)	
Lead Author	Enrico Mezzetti	
E-Mail	enrico.mezzetti@bsc.es	
Contributing Beneficiaries	DLR	• Henning Schlender
	IKL	• Victor Julio Ramirez • Ane Mijangos
	BSC	• Enrico Mezzetti • Javier Barrera • Francisco J. Cazorla
	STM FR	• Marcello Coppola
	ZSC	• Ivan Paez
	ITI	• Joan J. Valls • Miguel García • Javier Coronel
Reviewing Beneficiaries	AVL	• Hannes Fuchs
	DLR	Gregor Nitsche
Work Package(s)	1	
Related Task(s)	T1.1	Identification of node-level safety and security-relevant properties
Keywords	Software-Defined Vehicle, SDV, Horizon Europe, Automotive, CCAM, OTA, Update, Upgrade, DevOps, E/E-Architecture, Orchestration, Vehicle-to-cloud, Cloud-Edge Continuum, Reconfiguration, Safety properties, Security properties, SASE requirements	

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 (add and/or delete)
V0.1	01.12.2025	Initial version
V0.2	15.12.2025	Capturing first round of internal comments
V0.3	18.12.2025	Version for review
V0.4	05.01.2026	Reviewed Version
V1.0	05.01.2026	Final and reviewed Version (public)

Table of contents

Executive Summary	7
Acronyms and Abbreviations	8
List of Figures.....	9
List of Tables.....	9
References and Background Material	9
1. Introduction	11
1.1. Project Outline	11
1.2. Work-Package Outline	12
1.2.1. Work-Package Objectives	12
1.2.2. Work-Package Description	13
1.2.3. Work-Package Status.....	14
1.3. Purpose and Scope of the Deliverable	14
2. Identification of Safety and Security properties	15
2.1. Functional Safety dimension in UP2DATE4SDV	16
2.2. Security dimension in UP2DATE4SDV	17
3. UP2DATE4SDV SASE Properties	18
3.1. Intra-node Node Level Properties.....	21
3.1.1. SASE-Intra-01	21
3.1.2. SASE-Intra-02	21
3.1.3. SASE-Intra-03	22
3.1.4. SASE-Intra-04	22
3.1.5. SASE-Intra-05	23
3.1.6. SASE-Intra-06	23
3.1.7. SASE-Intra-07	24
3.1.8. SASE-Intra-08	25
3.1.9. SASE-Intra-09	25
3.1.10. SASE-Intra-10.....	26
3.1.11. SASE-Intra-11.....	27
3.1.12. SASE-Intra-12.....	27
3.1.13. SASE-Intra-13.....	28
3.1.14. SASE-Intra-14.....	28
3.1.15. SASE-Intra-15.....	29
3.1.16. SASE-Intra-16.....	30
3.1.17. SASE-Intra-17.....	30
3.1.18. SASE-Intra-18.....	31

3.1.19.	SASE-Intra-19.....	31
3.1.20.	SASE-Intra-20.....	32
3.1.21.	SASE-Intra-21.....	32
3.1.22.	SASE-Intra-22.....	33
3.1.23.	SASE-Intra-23.....	34
3.1.24.	SASE-Intra-24.....	34
3.1.25.	SASE-Intra-25.....	35
3.1.26.	SASE-Intra-26.....	36
3.1.27.	SASE-Intra-27.....	36
3.1.28.	SASE-Intra-28.....	37
3.1.29.	SASE-Intra-29.....	37
3.1.30.	SASE-Intra-30.....	38
3.1.31.	SASE-Intra-31.....	39
3.1.32.	SASE-Intra-32.....	39
3.1.33.	SASE-Intra-33.....	39
3.1.34.	SASE-Intra-34.....	40
3.1.35.	SASE-Intra-35.....	41
3.1.36.	SASE-Intra-36.....	41
3.1.37.	SASE-Intra-37.....	42
3.1.38.	SASE-Intra-38.....	42
3.1.39.	SASE-Intra-39.....	43
3.1.40.	SASE-Intra-40.....	44
3.1.41.	SASE-Intra-41.....	44
3.1.42.	SASE-Intra-42.....	45
3.1.43.	SASE-Intra-43.....	45
3.1.44.	SASE-Intra-44.....	46
3.2.	Inter-node Level Properties	47
3.2.1.	SASE-Inter-01.....	47
3.2.2.	SASE-Inter-02.....	47
3.2.3.	SASE-Inter-03.....	48
3.2.4.	SASE-Inter-04.....	48
3.3.	Alignment with other WPs.....	49
4.	Summary and Outlook	50

Executive Summary

This document fulfills the requirements for Deliverable D1.1 and reports on the efforts deployed in the scope of Task T1.1. for the identification of a set of relevant Safety and Security (SASE) properties for supporting safe and secure Over-The-Air (OTA) software updates, hardware upgrades and situation-dependent reconfigurations of software-defined vehicles (SDV).

The set of identified properties are meant to identify a high-level baseline of requirements to cooperate with other WPs in the definition of the project technological infrastructure and supported features. The set of properties are necessarily high-level as many of them can be provided through different mechanisms, depending on the support at the hardware and software level and on the overall strategy. Therefore, how those properties will be concretely met in the UP2DATE4SDV execution platform will be consolidated in the course of the project, ensuring consistency across the hardware, system software, middleware and orchestrator layers.

Acronyms and Abbreviations

Abbrev.	Full Name
AD	Autonomous-Driving
ADAS	Advanced Driver-Assistant System
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
D	Deliverable
Diss/Expl	Dissemination/Exploitation
DoA	Description of Action
DPC	Data Privacy Coordinator
EC	European Commission
ECU	Electronic Control Unit
EU	European Union
GA	Grant Agreement
HAL	Hardware Abstraction Layer
HiL	Hardware-in-the-Loop
HW	Hardware
IDS	Intruder Detection System
IP	Intellectual Property
JU	Joint Undertaking
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
ORP	Orchestration and Reconfiguration Plane
PaCo	Participant Contact
PC	Project Coordinator
PCM	Phase Change Memory
PCoCo	Primary Coordinator Contact
PGA	Project General Assembly
PPR	Periodic Project Report
PU	Public
R	Report
SASE	SAfety and SEcurity
SD-CCAM-V	Software-Defined CCAM vehicles

SDV	Software Defined Vehicle
SEN	Sensitive
SiL	Software-in-the-Loop
SoC	System-on-Chip
SoM	System-on-Module
SW	Software
TSN	Time Sensitive Network
UCL	Use Case Leader
V&V	Verification and validation
V2C	Vehicle to Cloud
WP	Work Package

List of Figures

Figure 1: Overview of the UP2DATE4SDV work-packages and their Interaction.	12
Figure 2 - Scoping of relevant SASE properties.....	15

List of Tables

Table 1: SASE Properties.	18
--------------------------------	----

References and Background Material

Internal Background

- [1] Project Proposal Project Part B: technical alignment (not public)

External Background

- [2] D4.1_UP2DATE-Middleware-foundations-V1.1.pdf: https://h2020up2date.eu/wp-content/uploads/2021/10/D4.1_UP2DATE-Middleware-foundations-V1.1.pdf
- [3] D3.2_ArchDefinition_v02.022.pdf: https://h2020up2date.eu/wp-content/uploads/2021/10/D3.2_ArchDefinition_v02.00.pdf
- [4] Federate Building Blocks (<https://github.com/CSA-FEDERATE/Proposed-BuildingBlocks>)
- [5] Eclipse SDV – SCORE Project Architecture (<https://eclipse-score.github.io/score/main/modules/index.html>)
- [6] HAL4SDV Platform Requirements (https://www.hal4sdv.eu/files/ugd/76d064_3352d3e51b5b43028f11254027a53b3a.pdf)
- [7] International Organization for Standardization. (2018). ISO 26262: Road vehicles — Functional safety [Standard]. <https://www.iso.org>.
- [8] International Organization for Standardization. Road vehicles — Safety of the intended functionality. ISO 21448:2022, 2022. <https://www.iso.org/standard/77490.html>

- [9] International Organization for Standardization (ISO) and the Society of Automotive Engineers (SAE) International. ISO/SAE 21434:2021, Road vehicles — Cybersecurity engineering. [Geneva, Switzerland]: ISO, 2021.

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 SDV.

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 connected cooperative and automated mobility (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 overarching 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 applicable 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**

The demonstrators and use-cases (developed in WP5) define requirements (pink arrows in Figure 1) 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 those requirements, 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 Model-Based System

Engineering (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 WPs and their communication, both internally and 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.

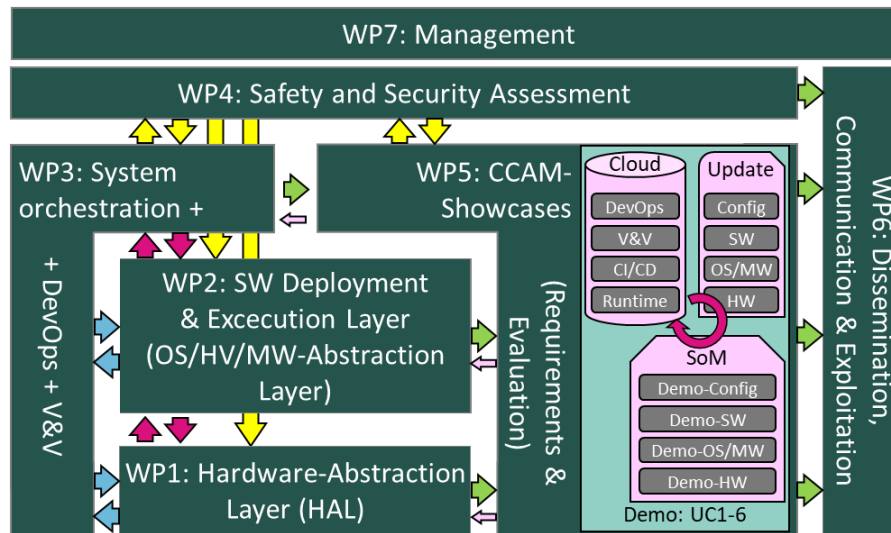


Figure 1: Overview of the UP2DATE4SDV work-packages and their Interaction.

1.2. Work-Package Outline

The overarching goal of UP2DATE4SDV, to enable continuous SASE-aware software and hardware updates and reconfiguration across complex, heterogeneous Commercial-Off-The-Shelf (COTS) hardware platforms, relies on dedicated support at both hardware and software levels. As shown in Figure 1, WP1 is a fundamental enabler for the overall project as it provides the necessary abstractions for supporting system-level monitoring and (re-)configuration objectives in the different layers in the UP2DATE4SDV execution platform. To this extent, WP1 focuses on both intra- and inter-node scopes and focuses on the necessary hardware-level features and functionalities to support UP2DATE4SDV objectives by working towards the definition of a modular and reusable HAL that guarantees the necessary decoupling between the execution platform and the functionalities it offers to the upper software layers. WP1 has direct interactions with WP2, for supporting the project SW-level abstractions, but also indirectly supports the way HAL functionalities are exposed and leveraged at the application level.

1.2.1. Work-Package Objectives

WP1 provides the low-level technological foundations to support the project's higher-level orchestration, deployment, safety, and security mechanisms. Its overarching objective is to design and implement a HAL capable of exposing reliable, fine-grained, and safety-/security-relevant observability and controllability features in COTS System-on-Chip (SoC) platforms. The WP moves from the identification of those properties in computing nodes that are essential to cover SASE aspects in CCAM systems supporting dynamic updates and reconfiguration, to enabling those properties in the project target platforms, under their specific supported features and limitations. WP1 also addresses the

design of monitoring services that extend beyond the SoC itself, capturing inter-node communication characteristics, which are indeed essential for supporting synchronized updates and distributed reconfiguration. Ultimately, WP1 also aims at providing a modular System-on-Module (SoM), integrating the target COTS devices, serving as a reference hardware platform for all other WPs.

1.2.2. Work-Package Description

WP1 is structured into five main tasks, each one covering a specific goal in the WP.

Task 1.1 Identification of node level safety and security-relevant properties. This task focuses on understanding SASE-relevant properties that characterize individual computing nodes but also extends to consider inter-node aspect to capture properties relevant for the interactions across the distributed system. The task is meant to identify all properties that influence SASE (safety and security) across hardware and communication layers, such as determinism, partitioning, redundancy, data integrity and access control, and to formalize them using appropriate metrics and model-based representations. SASE properties allow SW WPs (WP2–WP4) to reason coherently about the functionalities offered by the hardware layer.

Task 1.2 Identification of observability and controllability channels for the selected SoCs. It builds directly on T1.1 outcomes by identifying the actual observability and controllability capabilities of the selected COTS System-on-Chip platforms. The objective here is to examine in detail the concrete hardware knobs that can be leveraged to support monitoring and configurability. These knobs include: configuration registers, memory-mapped performance monitors, traffic counters, and other low-level SoC features. The task aims to theoretically (from available reference manuals) and empirically determine which concrete features can be reliably exposed through the HAL. The information collected and exposed at the hardware level can be integrated into the hypervisor control logic, the vehicle–cloud orchestration plane, and the safety/security mechanisms in software WPs.

Task 1.3 SoC configurations for safety and security-relevant processes. Based on the knobs identified in T1.2, this task aims to derive concrete SoC configuration profiles that support SASE-relevant processes for the system updates and reconfigurations. These configurations must enable secure, predictable, and fully controlled execution of software modules, both for isolated node activities and for distributed, multi-node update scenarios. Configurations are meant to remain compatible with the constraints imposed by the hypervisor and are amenable to model-based representations of resource availability and utilization for WP3 orchestrator.

Task 1.4 SoC and network support for monitoring services. This task extends the scope of WP1 beyond the SoC by focusing on the network layer, to ensure that monitoring capabilities are not restricted to local hardware behavior, but also encompass the timing, load, and reliability characteristics of inter-node communication. The task seeks to integrate SoC-level monitoring with network-level observability, offering a unified representation of execution, especially under deterministic communication schemes, to both the orchestrator and SASE mechanisms. In this way, the task ensures that the HAL provides not only node-centric information, but also system-wide insights essential for synchronized update strategies.

Task 1.5 SoM based modular COTS HW platform. This task focuses on the provision of a SoM hardware platform that consolidates the selected COTS devices into a modular and reusable unit for experimentation, integration, and validation. The idea is to provide a consistent reference platform on which the HAL and its observability and controllability services can be executed, evaluated, and

demonstrated across all technical work packages. The task aims to support early prototyping and later integration of more advanced WP1 outcomes on incremental releases.

1.2.3. Work-Package Status

The work package activities started in M1 at the beginning of the project. While T1.1 was the only active task in the first 3 months, all five tasks are active at the time of writing. The collaboration among WP partners has been ensured by monthly WP calls and on-demand calls among partners. The collection of SASE properties and requirements has occurred through the project collaboration portal. While properties and requirements can benefit from integration and refinement along the course of the project, WP1 progresses as planned and currently no delays are expected.

1.3. Purpose and Scope of the Deliverable

Deliverable D1.1, together with D2.1, D3.1 (and D5.1 in M3) belongs to the set of early (M6) project deliverables that are meant to consolidate the foundational concepts for the UP2DATE4SDV project. D1.1 aims at identifying a set of relevant properties for the target COTS SoCs platforms to support SASE updates and reconfiguration. Those properties, in turn, enable the definition of a set of requirements on the project HAL and on the interoperability of the latter with the upper SW stack. Requirements allow to scope the effort in the remaining tasks in the project, and guarantee a concrete mapping of the identified properties as a set of functionalities supported by the concrete hardware (and the lower SW interface, if need be).

The Deliverable is also fundamental to ensuring the requirements and related mechanisms are also translated into a representation that is amenable to model-based formalization (with corresponding metrics) and therefore exploited as part of the orchestrator logic.

Given the early stage of the project, we are also envisioning the identified requirements (and formalization thereof) could also be amended, tailored, and refined with the refinement of the interoperability of the diverse UP2DATE4SDV abstraction layers.

2. Identification of Safety and Security properties

WP1 follows a bottom-up, model-based approach to ensure that the hardware layer exposes standardized abstractions and metrics that covers higher layers' requirements in an effective and efficient way. To this purpose WP1 focuses first on the identification and classification of an extensive, platform-agnostic set of SASE-relevant hardware properties (e.g., partitioning, redundancy, timing determinism, integrity, access control) at both intra- and inter-node levels.

Those properties convey a set of high-level requirements on the different abstraction layers in the UP2DATE4SDV execution stack. Each property may need specific support from the HAL and System Software, including OS and Hypervisor support, to monitor critical configuration settings (that need to be preserved from inadvertent changes at operation) and relevant aspects of dynamic execution. The gathered information is meant to be exploited for the deployment of effective SASE mechanisms and SASE-informed system updates and reconfigurations.

In this early stage of the project, SASE properties are necessarily high-level as the concrete way to support them across UP2DATE4SDV execution stack depends on several aspects, including available HW and SW support, and overall design decisions for the particular SASE mechanism or orchestration plane deployed in the Demonstrators' setups (WP5). The strategy followed in T1.1 has been therefore focusing on trying to avoid any bias in the definition of the SASE properties by gathering a generic set of properties, independently from the expected HW/SW support in the execution platforms. On the one hand, we expect that not all those generic properties will be met in the concrete setups due to possibly insufficient support at the HW level, or incompatibilities in the system software layers. On the other hand, the requirements arising from the Demonstrators' setups may identify a subset of the identified SASE properties.

How SASE properties will be concretely met is therefore dependent on how those properties will be consolidated in the course of the project. It will be determined by an intersection (see Figure 2) among an ideal set of SASE properties, the Demonstrators' requirements, and the available HW/SW support. The continuous synchronization among WPs in UP2DATE4SDV will ensure consistency in how the final set of SASE properties are supported across the hardware, system software, middleware and orchestrator layers.

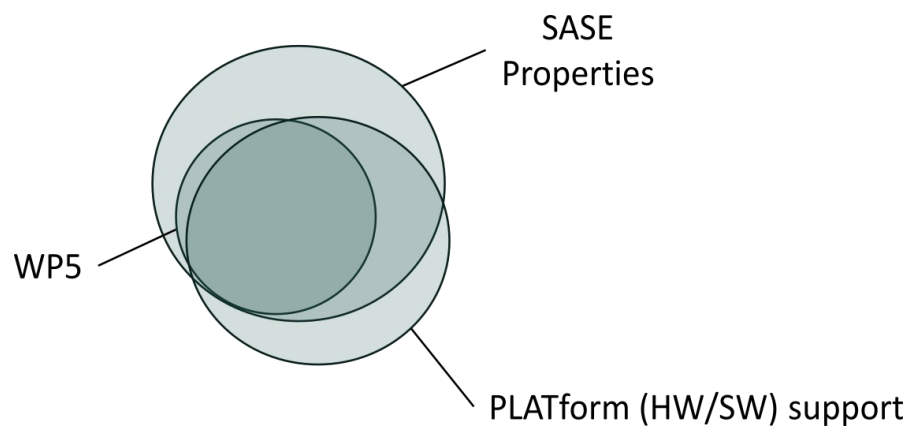


Figure 2 - Scoping of relevant SASE properties.

In the scope of WP1, the resulting set of SASE properties will be then mapped to concrete monitoring and configuration mechanisms available in modern heterogeneous COTS platforms selected for running the case studies and, as such, indirectly steer the definition of the HAL APIs for observability and controllability.

The HAL APIs will be then assessed against the requirements and specifications stemming from the upper layers (WP2-WP4), with which the HAL is meant to cooperate, and the concrete project use cases and demonstrators' setups (WP5).

By doing so, UP2DATE4SDV HAL ensures the alignment with the upper software layers and effectively enables the higher-level orchestration and update mechanisms to support safe, secure, and reliable system updates and reconfiguration, without exposing the underlying hardware complexities.

2.1. Functional Safety dimension in UP2DATE4SDV

Safety assurance for automotive E/E systems [7] depends on demonstrating that hardware and software elements operate deterministically, remain free from unintended interference, tolerate or detect faults, and maintain controlled behavior under both nominal and degraded conditions. At the same time, SOTIF (ISO/PAS 21448) [8] stresses the need to understand system behaviors arising from performance limitations, functional insufficiencies, or environmental conditions that may not be captured by traditional fault-based safety analysis. In the context of UP2DATE4SDV, effectively supporting OTA software updates, hardware upgrades, and run-time reconfigurations in modern CCAM and SDV architectures requires the underlying platform to exhibit certain safety characteristics or features. WP1, and in particular T1.1, addresses such requirement by identifying, formalizing, and structuring the safety-relevant properties of the compute nodes that constitute the foundation of the project's distributed execution platform.

Dynamic OTA updates, the introduction of new HW and SW modules, adaptive resource allocation, and coordinated multi-node update sequences modify the timing characteristics, execution context, and resource usage profile of software components, including safety-critical ones. For ISO 26262 compliance, such changes must be (i) evaluated against hardware properties such as execution determinism, spatial and temporal isolation, redundancy mechanisms, diagnostic coverage, to name a few, and (ii) steered by accurate hardware-level monitoring information. Similarly, in the scope of SOTIF, the system is expected to maintain an explicit, model-based understanding of node-level behavior and limitations, to timely react to performance degradation, uncertainty in sensor-driven computation, or the risk of insufficient. Task T1.1 provides this foundation by identifying a first set of hardware properties to be formalized by a model-based representation and to be tracked by measurable metrics and constraints that reflect the safety envelope within which updates and reconfigurations are meant to operate in a safety-critical context.

By capturing these properties in a structured, machine-readable form, WP1 enables SW to incorporate safety-aware reasoning into their mechanisms. The more obvious interdependency is with WP4, where SASE mechanisms are designed and developed: SASE properties are fundamental for the SASE assessment and the deployment of SASE mechanisms, ensuring that safety analyses reflect both classical ISO 26262 fault models and SOTIF's broader behavioral concerns. Besides WP4, all upper abstractions leverage SASE properties. WP2 can use the SASE property models to enforce strict isolation and determinism through the hypervisor. WP3 relies on SASE properties to ensure that orchestration and scheduling decisions do not violate safety constraints when deploying or updating software components.

Ultimately, bringing functional safety principles in the HAL, contributes to the definition of a safety argument in which hardware characteristics, monitoring capabilities, and configuration constraints are explicitly linked to project-wide safety requirements.

2.2. Security dimension in UP2DATE4SDV

Modern in-vehicle computing nodes operate in highly connected environments, interface with external ecosystems, and execute critical functions whose correctness and availability must be preserved in the presence of malicious threats. This is even more critical during system updates and re-configuration as the system is inherently more fragile to attacks and malicious threats. In alignment with ISO/SAE 21434 [9], all threat scenarios targeting communication interfaces, software-update mechanisms, computation resources, timing sources, or memory and storage subsystems may compromise the confidentiality, integrity, availability, and authenticity of in-vehicle functions. Such cybersecurity events can propagate to safety-critical domains and thereby contribute to hazardous behavior.

Security is a prerequisite for functional safety and, therefore, a fundamental dimension in UP2DATE4SDV and security is explicitly addressed as a core pillar in the SASE-oriented solution, together with functional safety. While the specific SASE mechanisms and update/reconfiguration frameworks are addressed in other WPs, Task T1.1 (and WP1 in general) is responsible for the identification and early formalization of the hardware- and software-level properties that contribute to the secure operation, update, and configuration of each computing node. The set of SASE properties identified in this deliverable is meant to identify the building blocks for capturing security-level concerns. This includes secure boot and attestation capabilities, protection of in-vehicle and external communication, isolation and segregation mechanisms, hardware monitoring functions, memory integrity features, and the set of security-relevant configuration options and observability channels.

However, similarly to functional safety properties, security properties require specific support. While T1.1 establishes the conceptual taxonomy and defines the set of security properties at an abstract level, the feasibility, granularity, and enforceability of these properties are determined by the actual support provided by the underlying platform. For instance, mechanisms such as secure boot, attestation, hardware isolation, fault-detection monitors, memory-integrity features, secure communication accelerators, or runtime observability counters require explicit hardware primitives and operating-system or hypervisor services to be functional in practice. These platform-specific aspects are only partially anticipated in this report and are meant to be analyzed and consolidated in the subsequent WP1 tasks, defining how SASE properties can be exposed and configured through the concrete APIs and monitoring.

By establishing a comprehensive taxonomy of these security properties, WP1 T1.1 provides the foundation upon which secure orchestration (WP3) and secure update and validation workflows (WP4) are built, ensuring that the system remains within a trusted operational envelope throughout its lifecycle.

3. UP2DATE4SDV SASE Properties

In the following we provide the first set of 48 high-level SASE properties identified in T1.1. The properties capture aspects at the level of computing nodes (*intra-node properties*) but also extend to those node aspects that are relevant for communication between nodes (*inter-node properties*). The individual entries have been shared within the consortium through the *Confluence* interface, to foster collaboration and optimize the feedback loop. Table 1 below provides a summary of intra- and inter-node properties, linking to a specific entry for each property where we report:

- **Property ID:** unique identifier for the SASE property
- **Relevance:** whether the property is relevant for the Safety and Security dimension
- **Related properties:** other conceptually related properties
- **Scope:** whether the property is exclusively at node level or involves communication
- **Level:** whether the property is mainly enabled at HW or SW level
- **Type:** whether the property is Mandatory (Must), Desirable (Should), or Optional (Could)
- **Description:** short text describing the property
- **Rationale:** explanation of why and how the property is relevant
- **Monitoring:** monitoring aspects to support the enforcement or assessment of the property
- **Controllability:** configuration aspects related to the property
- **Metrics:** example metrics related to the property
- **Formalization:** whether the property requires formalization
- **WPs:** work packages supporting or exploiting the property

Table 1: SASE Properties.

ID	Property
SASE-Intra-01	Hardware support for segregation on the shared cache data array among cores
SASE-Intra-02	Hardware support for segregation on the shared cache request buffer partitioning among cores
SASE-Intra-03	Hardware support to allow several requests to be serviced at the same time from the interconnect (e.g. crossbar)
SASE-Intra-04	Hardware support for disabling caching in main interconnect (if caching available)
SASE-Intra-05	Hardware-level support of QoS for the requests sent by different sources to the interconnect (cores, accelerators, etc)
SASE-Intra-06	Hardware-level support to allow accelerators to be shared in a deterministic way among kernels from different applications
SASE-Intra-07	Hardware-level support to assign different compute/memory/interconnect blocks of the accelerator to different kernels
SASE-Intra-08	Hardware support for splitting the buffers, queues, and other resources within the memory controller shared among cores and computing elements

SASE-Intra-09	Hardware support for assigning different levels of priorities among requests sent to the memory controller from different cores or accelerators
SASE-Intra-10	Hardware support to prevent or significantly limit the reordering of memory requests that the hardware is allowed to do
SASE-Intra-11	Hardware support to assign the requests from different cores/accelerators to different memory controllers
SASE-Intra-12	Hardware support to prevent the core frequency and voltage from being automatically (inadvertently) changed
SASE-Intra-13	Hardware support to change the processor (MPSoC) power mode under which the hardware is allowed to operate
SASE-Intra-14	Hardware support for redundancy and diversity in executing a given software task
SASE-Intra-15	Hardware support to ensure data integrity (ECC in caches and Flash memory, EDC in RAM, and CRC in communications, CRC unit)
SASE-Intra-16	Hardware support for flushing stateful hardware resources (caches, buffers, BTB, etc)
SASE-Intra-17	Software support from the OS or Hypervisor to allow binding tasks to specific computing cores
SASE-Intra-18	Software support for the accelerator driver to place kernels in particular partitions of the accelerator
SASE-Intra-19	Software support from the OS to allow binding IRQ to a specific computing core
SASE-Intra-20	Software support to bind non-critical OS/hypervisor activities to the desired core(s)
SASE-Intra-21	Hardware support for providing a secure area inside a device's main processor that ensures sensitive data and code are protected (Intel SGX, ARM TrustZone, etc)
SASE-Intra-22	Hardware and Software support to secure boot (UEFI)
SASE-Intra-23	Hardware and Software support for encrypting sensitive data, both at rest and in operation
SASE-Intra-24	Hardware and Software level support for secure firmware updates (secure bootloader, failsafe mechanism during firmware updates, etc.)
SASE-Intra-25	Software-level support to minimize disruption and support automatic/on-demand updates and patches
SASE-Intra-26	Software support for a monitoring strategy aiming at intercepting and reacting to suspicious events and patterns (logs should also be protected from external manipulation)
SASE-Intra-27	Hardware and software support to protect the System from physical threats (e.g., access)
SASE-Intra-28	Hardware and software support to enable the system to react and recover from safety and security failures with minimal disruption of critical services
SASE-Intra-29	The system (HW and SW) minimizes the attack surface by removing unnecessary components and ensuring the system is up to date
SASE-Intra-30	Hardware support for a watchdog with separate time-base and time-window
SASE-Intra-31	Hardware support for monitoring the power supply's output
SASE-Intra-32	Hardware shall support a Clock Supervisor capable of detecting anomalies in program execution sequences caused by timing faults or clock malfunctions
SASE-Intra-33	Hardware support to power-down with safety shut-off in case of overvoltage or undervoltage so that those scenarios are promptly detected and, when necessary, the internal state can be saved in non-volatile memory

SASE-Intra-34	Software support to isolate update execution per partition with enforced Security Levels and communication policies
SASE-Intra-35	Software support to monitor runtime behavior and securely log events for audit and anomaly detection
SASE-Intra-36	Hardware support for identifying the number and type of active computing elements
SASE-Intra-37	Hardware support for monitoring the load of different computing elements
SASE-Intra-38	Hardware support for tracking the temperature in each computing element
SASE-Intra-39	Hardware support for monitoring CPU cache behavior per core and per cache level
SASE-Intra-40	Hardware support for retrieving the different types of memories available in the system and the number of units per type
SASE-Intra-41	Hardware support for gathering information of the memory bandwidth utilization
SASE-Intra-42	Hardware support for gathering the number of memory faults experienced by different memories, per type and id
SASE-Intra-43	Hardware support for gather metrics regarding the utilization (percentage) of the storage capabilities of each memory by type and id
SASE-Intra-44	The hardware platform should support the replacement or addition of SoM elements or alike while preserving system functionality, safety, security, and performance constraints
SASE-Inter-01	Hardware and Software support to secure in-vehicle and external communication
SASE-Inter-02	Hardware and software support to obtain inter-nodes communication logs (these logs are required to build datasets, which serve as a crucial input for the training of Intruder Detection System (IDS))
SASE-Inter-03	Hardware and software support for enabling real-time packet capturing and monitoring of inter-node communications from a specific node for intrusion detection purposes
SASE-Inter-04	Hardware and software support for capturing the bandwidth of the network interface

3.1. Intra-node Node Level Properties

In the following, we provide an identification and description of node-level SASE properties.

3.1.1. SASE-Intra-01

Property ID	SASE-Intra-01
Relevance	Safety Security
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support for segregation on the shared cache data array among cores.
Rationale	Applying segregation on the shared cache data array allows reserving a private space of the data array for each core, providing predictable execution to each core by protecting its data from being evicted by a contender (timing dimension) or tampered with (security dimension). The capability to support partitioned data caches directly affects freedom from interference, timing determinism, data integrity, and resilience to security attacks (shared caches are one of the most common sources of side-channel attacks on multicore SoCs). ISO 26262-6:2018 (Annex D) and ISO 26262-9 mention cache partitioning or disabling as interference avoidance and mitigation techniques. Partitioning also captures ISO/SAE 21434 requirement for reducing exploitable attack surfaces.
Monitoring	Yes. Critical configuration setting i. Cache arrays segregation status: [unsupported, enabled, disabled] ii. <Data array element, core> mapping
Controllability	i. Configuration register (may require kernel module, kernel capabilities)
Metrics	i. Cache partitions count and %
Formalization	TBC
WPs	WP1, WP4

3.1.2. SASE-Intra-02

Property ID	SASE-Intra-02
Relevance	Safety Security
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support for segregation on the shared cache request buffer partitioning among cores.
Rationale	Providing segregation capabilities on the shared cache request buffers allows reserving a private space of the buffer for each core, reducing execution time variability by avoiding resource contention between cores and reducing the attack surface. Similarly to data cache partitioning, capability to support partitioned buffers directly affects freedom from interference, timing determinism, data integrity, and resilience to security attacks. ISO 26262-

	6:2018 (Annex D) and ISO 26262-9 mention partitioning and avoidance of shared resources as interference avoidance and mitigation techniques. Partitioning also captures ISO/SAE 21434 requirement for reducing exploitable attack surfaces.
Monitoring	Yes. Critical configuration setting i. Cache buffers segregation status: [unsupported, enabled, disabled] ii. < Cache buffer element, core > mapping
Controllability	i. Configuration register (may require kernel module, kernel capabilities)
Metrics	i. Cache buffer partitions count and %
Formalization	TBC
WPs	WP1, WP4

3.1.3. SASE-Intra-03

Property ID	SASE-Intra-03
Relevance	Safety
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support to allow several requests to be serviced at the same time from the interconnect (e.g. crossbar).
Rationale	Providing support for several parallel requests avoids delaying the transmission of requests, as they no longer contend for the use of the interconnection, thus, potentially increasing individual performance and reducing execution time variability. This is typically an inherent feature for an interconnect (e.g., crossbar) but may also be a configurable setting.
Monitoring	-
Controllability	Typically, inherent HW feature (may be parametrizable in some cases) i. Number of requests in parallel supported: [1,MAX]
Metrics	-
Formalization	-
WPs	WP1, WP4

3.1.4. SASE-Intra-04

Property ID	SASE-Intra-04
Relevance	Safety Security
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support for disabling caching in main interconnect (if caching available).
Rationale	Cache in the interconnect exposes applications to interference from other devices (timing dimension) and/or malicious attacks (security dimension). It is important to understand if caching is implemented at the interconnect level and possibly be able to disable to reduce sources of variability and minimize

	the attack surface. Same rationale as per cache applies: avoidance of shared resources prevents timing interferences (ISO 26262-6:2018 (Annex D) and ISO 26262-9) and reduces the exploitable attack surface (ISO/SAE 21434).
Monitoring	Yes. Critical configuration setting i. Interconnect cacheability status: [unsupported, enabled, disabled]
Controllability	May need explicit enabling/disabling in some cases i. Configuration register (may require kernel module, kernel capabilities)
Metrics	-
Formalization	-
WPs	WP1, WP4

3.1.5. SASE-Intra-05

Property ID	SASE-Intra-05
Relevance	Safety
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware level support of QoS for the requests sent by different sources to the interconnect (cores, accelerators, etc).
Rationale	Hardware-level QoS mechanisms on the SoC interconnect determine how memory, peripherals, accelerators, and other shared resources are accessed by different hardware components (CPUs, DMAs, GPUs, NPUs, etc.). As the SoC interconnect is one of the highest-impact sources of interference, enabling different levels of QoS for interconnect requests allows for determining service guarantee for high-criticality functions, hence reducing the impact of interference and improving time determinism. Needless to say, that adequate QoS setting may also prevent security concerns, such as for example, denial of service attacks, starvation of critical tasks, or inferring system activity from bus-level timing variations.
Monitoring	Yes. Critical configuration setting i. Interconnect QoS prioritization support: [unsupported, enabled, disabled]
Controllability	May need explicit enabling/disabling in some cases i. < Computing element, QoS class > (may require kernel module, kernel capabilities)
Metrics	Metrics may be defined based on the specific component and QoS mechanism
Formalization	-
WPs	WP1, WP3, WP4

3.1.6. SASE-Intra-06

Property ID	SASE-Intra-06
Relevance	Safety Security

Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware level support to allow accelerators to be shared in a deterministic way among kernels from different applications.
Rationale	The use of accelerators is a critical enabler for perception tasks, AI inference, signal processing, especially when they contribute to safety-critical functions. Accelerator hardware blocks represent critical shared resources that are concurrently accessed by multiple applications, with diverse criticality levels. Being able to share these accelerators deterministically, with bounded latency and freedom from interference, is therefore key for functional safety goals. Deterministic sharing is also instrumental to prevent SOTIF performance degradation in AI/perception workloads. From the security perspective, enforcing strict deterministic behavior ensures similar results as partitioning, allowing the mitigation of diverse cybersecurity risks such as denial-of-service or timing side-channel attacks.
Monitoring	Yes. Critical configuration setting i. Deterministic sharing support [unsupported, enabled, disabled]
Controllability	May need explicit enabling/disabling in some cases i. Depending on the available mechanism (may require kernel module, kernel capabilities)
Metrics	Metrics may be defined based on the specific component and QoS mechanism
Formalization	-
WPs	WP1, WP4

3.1.7. SASE-Intra-07

Property ID	SASE-Intra-07
Relevance	Safety Security
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware-level support to assign different compute/memory/interconnect blocks of the accelerator to different kernels.
Rationale	Accelerator hardware blocks represent critical shared resources that are concurrently accessed by multiple applications, with diverse criticality levels. Being able to enforce fine-grained assignments of sub-blocks in those accelerators (compute/memory/interconnect elements) instrumental to guarantee freedom from interference and prevent SOTIF performance degradation in AI/perception workloads. Again, also from the security perspective, partitioning allows for the mitigation of diverse cybersecurity risks such as denial-of-service or timing side-channel attacks.
Monitoring	Yes. Critical configuration setting i. Partitioning support [unsupported, enabled, disabled]
Controllability	May need explicit enabling/disabling in some cases

	i. < Hardware partition, core > mapping (may require kernel module, kernel capabilities)
Metrics	i. Partitions count and %
Formalization	-
WPs	WP1, WP4

3.1.8. SASE-Intra-08

Property ID	SASE-Intra-08
Relevance	Safety Security
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support for splitting the buffers, queues, and other resources within the memory controller among cores and computing elements.
Rationale	Similarly to caches and buffers, being able to enforce fine-grained assignments of resources inside the memory controllers allows improving execution time determinism and reduce the sources of timing interference (ISO 26262-6:2018 (Annex D) and ISO 26262-9) as well as prevent SOTIF performance degradation in AI/perception workloads. Again, also from the security perspective, partitioning of sub-blocks inside the memory controller allows for the mitigation of diverse cybersecurity risks: prevents denial-of-service attacks, eliminates timing and contention-based side channels, isolates trusted and untrusted execution domains, and ensures that a compromised core cannot disrupt or observe other workloads.
Monitoring	Yes. Critical configuration setting i. Intra-memory controller resource partitioning: [unsupported, enabled, disabled]
Controllability	May need explicit enabling/disabling in some cases i. < Memory controller partition, core > mapping (may require kernel module, kernel capabilities)
Metrics	i. Partitions count and %
Formalization	-
WPs	WP1, WP4

3.1.9. SASE-Intra-09

Property ID	SASE-Intra-09
Relevance	Safety
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support for assigning different levels of priorities among requests sent to the memory controller from different cores or accelerators.
Rationale	Hardware mechanisms that allow assigning different priority levels in the memory controller to master components (via priority arbitration, AXI QoS

	fields, traffic classes, credit systems, time division, etc.) directly influence timing guarantees, segregation among mixed-criticality applications, and robustness against timing interference.
Monitoring	Yes. Critical configuration setting i. Memory controller QoS prioritization support: [unsupported, enabled, disabled]
Controllability	i. < Computing element, QoS class> mapping (may require kernel module, kernel capabilities)
Metrics	Metrics may be defined based on the bandwidth or stalls depending on the specific component and QoS mechanism
Formalization	-
WPs	WP1, WP4

3.1.10. SASE-Intra-10

Property ID	SASE-Intra-10
Relevance	Safety
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support to prevent or significantly limit the reordering of memory requests that the hardware is allowed to do.
Rationale	Modern multicore SoCs aggressively reorder memory requests (loads/stores, DMA bursts, accelerator accesses) to optimize throughput. However, this reordering introduces nondeterministic latency, timing interference, and potential information flows, which directly impacts safety, performance trustworthiness (and possibly also security). Memory-request reordering may break time determinism by exposing load and stores to unpredictable delays, and allows interference among tasks to arise due to shared ordering buffers. Hardware mechanisms that disable, or regulate memory-request reordering (e.g., via ordering fences, strict-consistency channels, deterministic interconnect modes, and globally observed ordering rules) are therefore required to support safety goals.
Monitoring	Yes. Critical configuration setting i. Memory requests reordering: [unsupported, enabled, disabled]
Controllability	May need explicit enabling/disabling in some cases i. Depending on the mechanism may require kernel module, kernel capabilities
Metrics	-
Formalization	-
WPs	WP1, WP4

3.1.11. SASE-Intra-11

Property ID	SASE-Intra-11
Relevance	Safety Security
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support to assign the requests from different cores/accelerators to different memory controllers.
Rationale	Modern SoCs often include multiple memory controllers, each with their own queues, buffers, bandwidth paths, and arbitration logic. Hardware support that allows directing the requests of different hardware components (CPUs, DMAs, NPUs, GPUs, DSPs) to different memory controllers enables isolation, timing determinism, performance sufficiency, and attack-resilience. Assigning requests from different cores to different memory controllers ensures temporal and spatial isolation of memory traffic (favoring predictable and bounded latency) and therefore stable WCET under high load or during updates. From the security standpoint, partitioning the memory controllers may help preventing starvation of critical workloads as well as leakage of timing-based information.
Monitoring	Yes. Critical configuration setting i. Computing element to Memory controller binding: [unsupported, enabled, disabled]
Controllability	May need explicit enabling/disabling in some cases i. < Memory controller partition, computing device> mapping (may require kernel module, kernel capabilities)
Metrics	i. Controllers count and % allocation
Formalization	-
WPs	WP1, WP3, WP4

3.1.12. SASE-Intra-12

Property ID	SASE-Intra-12
Relevance	Safety
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support to prevent the core frequency and voltage from being automatically (inadvertently) changed.
Rationale	Modern SoCs include mechanisms such as DVFS (Dynamic Voltage and Frequency Scaling), power-saving governors, thermal throttling, and automated boost modes. These mechanisms may end up heavily affecting execution time determinism. Indeed, not all safety goals can be guaranteed under all operating conditions. Ensuring that frequencies (for each computing element) and voltage remains as configured helps avoiding unwanted variabilities and peaks in both the timing and energy profiles.

Monitoring	Yes. Critical configuration setting i. Frequency and voltage lock: [unsupported, enabled, disabled]
Controllability	May need explicit enabling/disabling in some cases i. Depending on the mechanism may require kernel module, kernel capabilities
Metrics	i. Frequency (per computing element or cluster thereof) ii. Voltage
Formalization	-
WPs	WP1, WP4

3.1.13. SASE-Intra-13

Property ID	SASE-Intra-13
Relevance	Safety Security
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support to change the power mode under which the hardware is allowed to operate.
Rationale	Hardware support for (controlled) transitioning between different power modes is a SASE property because both functional safety (ISO 26262 / SOTIF) and cybersecurity critically depend on the system's ability to manage, monitor, and control how hardware components behave under changing power and performance conditions, especially during OTA updates, run-time reconfigurations, and fail-operational scenarios. From a cybersecurity and system integrity perspective, power-mode control is also a security property: attackers could potentially exploit uncontrolled power-mode transitions to downgrade a system into a vulnerable operating point, disable certain IPs, reduce diagnostic capabilities, or impair cryptographic modules.
Monitoring	Yes. Critical configuration setting i. Power mode change: [unsupported, enabled, disabled]
Controllability	i. Depending on the mechanism may require kernel module, kernel capabilities
Metrics	i. Current power mode (and performance profile)
Formalization	-
WPs	WP1, WP3, WP4

3.1.14. SASE-Intra-14

Property ID	SASE-Intra-14
Relevance	Safety
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable

Description	Hardware support for redundancy and diversity in executing a given software task
Rationale	Diversity and redundancy in the execution of a task are SASE properties because they contribute directly to fault tolerance, fail-operational behavior, attack resilience, and predictable system recovery, which are an inescapable requirement for safe and secure updates and run-time reconfigurations. From a safety perspective, diversity and redundancy enable fault detection (cross-checking between channels), fault tolerance (backup execution paths), fail-operational behavior, reduction of common-mode failures, ASIL decomposition (by demonstrating independent execution channels). Multiple redundancy and diversity mechanisms are possible (e.g., dual-core lockstep, redundant accelerators with comparator logic, etc.)
Monitoring	Yes. Critical configuration setting i. Hardware component redundancy: [unsupported, enabled, disabled]
Controllability	May need explicit enabling/disabling in some cases i. Depending on the mechanism may require kernel module, kernel capabilities
Metrics	Metrics may be defined based on the diversity and redundancy scheme and mechanism
Formalization	-
WPs	WP1, WP4

3.1.15. SASE-Intra-15

Property ID	SASE-Intra-15
Relevance	Safety Security
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support to ensure data integrity (ECC in caches and Flash memory, EDC in RAM, and CRC in communications, CRC unit)
Rationale	Correctness and reliability of data are fundamental prerequisites for functional safety (ISO 26262), SOTIF-compliant behavior (ISO/PAS 21448), and cybersecurity (ISO/SAE 21434). Ensuring data integrity is therefore a critical safety and security goal and prevents the system from entering unsafe or untrusted states during operation, updates, or reconfiguration. Both orchestration (WP3) and SASE assessment (WP4) rely on guarantees that data is not corrupted when performing update sequences, migrating tasks, or validating safety envelopes. Loss of data integrity during an update can lead to incorrect deployment, unsafe behavior, or inconsistent system states. Integrity mechanisms such as ECC, parity checks, hardware error detection, and trusted memory protection prevent silent corruption, detect tampering, and preserve the validity of safety and security critical data both at operation and upon system updates
Monitoring	Yes. Critical configuration setting i. Support for ECC/EDC in memory device: [unsupported, enabled, disabled] ii. CRC unit: [unsupported, enabled, disabled]

Controllability	May need explicit enabling/disabling in some cases i. Depending on the mechanism may require kernel module, kernel capabilities
Metrics	Metrics may be defined based on the diversity and redundancy scheme and mechanism
Formalization	-
WPs	WP1, WP4

3.1.16. SASE-Intra-16

Property ID	SASE-Intra-16
Relevance	Safety
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support for flushing stateful hardware resources (caches, buffers, BTB, etc)
Rationale	Flushing stateful hardware resources is a useful way to remove undesired track record of certain hardware structures. Resetting state before running safety tasks prevents cross-contamination and non-determinism in execution time, especially when switching between different criticalities. Flushing is also fundamental to support OTA updates and reconfiguration as the system is meant to execute from a clean state.
Monitoring	-
Controllability	ISA support. If based on memory-mapped register may require kernel module, kernel capabilities.
Metrics	i. Count of flush event for a given resource
Formalization	-
WP	WP1, WP3, WP4

3.1.17. SASE-Intra-17

Property ID	SASE-Intra-17
Relevance	Safety
Related properties	N/A
Scope	Intra-node
Level	SW
Type	Desirable
Description	Software support from the OS or Hypervisor to allow binding tasks to specific computing cores
Rationale	Binding a task to a specific core (also called CPU affinity, core pinning, or explicit scheduling partitioning) gives the OS or hypervisor the ability to contribute to deterministic execution. In particular, controlling how tasks/processes are mapped to computing elements can be leveraged to enforce separation of software with different ASIL levels, and to guarantee computing resource availability at run time. These aspects are fundamental for guaranteeing predictable behavior in mixed-criticality systems and during

	OTA updates. Even though affinity control is implemented in software (OS/hypervisor), it is tightly dependent on hardware topology, core types, shared resources
Monitoring	May be implemented at SW level for relevant events
Controllability	i. <tasks, core> mapping, requires OS/Hypervisor level support
Metrics	i. Number of task migrations ii. Number of preemptions iii. Core task load (%)
Formalization	-
WPs	WP1, WP2, WP3, WP4

3.1.18. SASE-Intra-18

Property ID	SASE-Intra-18
Relevance	Safety
Related properties	N/A
Scope	Intra-node
Level	SW
Type	Desirable
Description	Software support for the accelerator driver to place kernels in particular partitions of the accelerator
Rationale	Many modern accelerators support hardware partitions, contexts, execution islands, or compute slices. These partitions provide spatial or temporal isolation across multiple applications, some of which may be safety-critical or even security-sensitive. Fine-grained, driver-level, software support to control how kernels are bounded to particular accelerator partitions is fundamental to maintaining control on how hardware resources are shared among different cores to guarantee timing determinism, ASIL-compliant isolation, and freedom from interference even in the accelerators. Similarly to task to core mapping, kernel mapping can be leveraged to enforce separation of software with different ASIL levels, and guarantee computing resource availability at run time. Even though affinity control is implemented in software (OS/hypervisor), it is tightly dependent on hardware-level support
Monitoring	May be implemented at SW level for relevant events
Controllability	i. <kernels, accelerator partitions> mapping, requires library support
Metrics	i. Number of kernel migrations ii. Number of kernel preemptions iii. Accelerator partition load (%)
Formalization	-
WPs	WP1, WP2, WP3, WP4

3.1.19. SASE-Intra-19

Property ID	SASE-Intra-19
Relevance	Safety
Related properties	N/A

Scope	Intra-node
Level	SW
Type	Desirable
Description	Software support from the OS to allow binding IRQ to a specific computing core
Rationale	Allowing the binding of IRQ to a specific computing core is fundamental to isolate critical tasks from unintended impact coming from interrupt handlers. Interrupt handlers' unpredictable activation patterns and variable latency hamper deterministic execution and performance guarantee. Even though affinity control is implemented in software (OS/hypervisor), it is tightly dependent on hardware-level support
Monitoring	May be implemented at SW level for relevant events
Controllability	i. <IRQ, core> mapping, requires OS/Hypervisor level support
Metrics	i. Number of IRQs processed per core
Formalization	-
WPs	WP1, WP2, WP3, WP4

3.1.20. SASE-Intra-20

Property ID	SASE-Intra-20
Relevance	Safety
Related properties	N/A
Scope	Intra-node
Level	SW
Type	Desirable
Description	Software support to bind non-critical OS/hypervisor activities to the desired core(s)
Rationale	Besides binding application tasks and IRQs, in general, binding non-critical OS threads, hypervisor management tasks, background services, daemons, and maintenance functions are equally critical to functional safety. If these low-criticality activities are allowed to run on arbitrary cores, they can degrade or interfere with the execution of safety-critical or security-critical tasks. In the same way as for IRQs, the binding of non-critical OS/hypervisor activities to a desired core or set thereof allows relegating those non-critical activities to dedicated core or where low criticality functions are executed. Such binding is useful in order to avoid the undesired interference from OS/Hypervisor on critical tasks. Even though affinity control is implemented in software (OS/hypervisor), it is tightly dependent on hardware-level support
Monitoring	May be implemented at SW level for relevant events
Controllability	i. <OS/Hypervisor activities, cores> mapping, requires OS/Hypervisor level support
Metrics	i. Number of non-critical processes running on cores reserved for critical tasks
Formalization	-
WPs	WP1, WP2, WP3, WP4

3.1.21. SASE-Intra-21

Property ID	SASE-Intra-21
--------------------	---------------

Relevance	Security
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Mandatory
Description	Hardware support for providing a secure area inside a device's main processor that ensures sensitive data and code are protected (Intel SGX, ARM TrustZone, etc)
Rationale	Technologies like Intel SGX, Arm TrustZone, RISC-V Keystone, and hardware-backed secure enclaves create an isolated execution environment within the processor where sensitive code can run securely, data is accessible only within the enclave, external software (OS, hypervisor, drivers) cannot tamper with or observe execution, and cryptographic keys and update metadata remain protected. Trusted execution environments provide foundational confidentiality, integrity, and authenticity mechanisms. Having a secure area inside the main processor is fundamental for protecting critical tasks from attackers, as well as keeping data integrity. It also enables the capability to securely boot and perform updates, as only trusted code is located in that secure area. Isolating secure-sensitive activities allows for an easier analysis of the safety and security of critical tasks
Monitoring	i. Secure area support: [unsupported, enabled, disabled] ii. Secure area status: [secure, violated]
Controllability	May need explicit enabling/disabling/configuration in some cases i. <kernel, trusted/untrusted> mapping
Metrics	i. Number of detected violations/violation attempts
Formalization	-
WPs	WP1, WP3, WP4

3.1.22. SASE-Intra-22

Property ID	SASE-Intra-22
Relevance	Security
Related properties	N/A
Scope	Intra-node
Level	HW and SW
Type	Desirable
Description	Hardware and Software support to secure boot (UEFI)
Rationale	Secure boot ensures that only authenticated, trusted code executes at every stage of the boot chain and is therefore a fundamental prerequisite for trustworthiness, isolation, update security, and integrity of safety-critical software. Secure boot establishes the root of trust for the entire execution platform, ensuring that the system always starts from a trusted, authenticated, and uncompromised state, hence fulfilling cybersecurity requirements of ISO/SAE 21434. It also supports ISO 26262 safety goals by guaranteeing startup integrity
Monitoring	i. Secure boot support: [unsupported, enabled, disabled]
Controllability	May need explicit enabling/disabling/configuration in some cases, possibly requiring OS/Hypervisor level support

Metrics	i. Attestation signals (e.g. SECURE_BOOT_OK, IMAGE_AUTH_PASS / FAIL, TRUSTZONE_ENABLED)
Formalization	-
WPs	WP1, WP4

3.1.23. SASE-Intra-23

Property ID	SASE-Intra-23
Relevance	Security
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware and Software support for encrypting sensitive data, both at rest and in operation
Rationale	Encryption capabilities (e.g., hardware-backed encryption engines, secure key storage, and protected memory regions) are fundamental enabler to as they contribute to protect the confidentiality and integrity of safety- and security-critical information, prevents tampering with safety-related configurations, and ensures that the platform operates in a trustworthy, predictable state
Monitoring	i. Data encryption support: [unsupported, enabled, disabled]
Controllability	May need explicit enabling/disabling/configuration in some cases, possibly requiring OS/Hypervisor level support
Metrics	-
Formalization	-
WPs	WP1, WP4

3.1.24. SASE-Intra-24

Property ID	SASE-Intra-24
Relevance	Security
Related properties	N/A
Scope	Intra-node
Level	HW and SW
Type	Mandatory
Description	Hardware and Software level support for secure firmware updates (secure bootloader, failsafe mechanism during firmware updates, etc.)
Rationale	Firmware updates modify the critical software components that ensure the safety and security of the system. The entire update pipeline must therefore be trusted, robust, and protected from attacks. Several mechanisms may be deployed and cooperate to ensure correct and secure firmware updates, providing: integrity of the update package, rollback and anti-replay protection, protection of update keys and credentials, etc. Secure update mechanisms directly support the cybersecurity requirements of ISO/SAE 21434, and also ensure startup integrity and safe reconfiguration required by ISO 26262
Monitoring	For each specific mechanism: i. Mechanism: [unsupported, enabled, disabled]

Controllability	May need explicit enabling/disabling/configuration in middleware, possibly requiring OS/Hypervisor level support
Metrics	Metrics may be defined based on the diversity and redundancy scheme and mechanism
Formalization	-
WPs	WP1, WP3, WP4

3.1.25. SASE-Intra-25

Property ID	SASE-Intra-25
Relevance	Safety Security
Related properties	N/A
Scope	Intra-node
Level	SW
Type	Desirable
Description	Software-level support to minimize disruption and support automatic/on-demand updates and patches
Rationale	While this property is meant to be implemented at the software/services/orchestration layers, it directly affects the security dimension in the computing node. On the one hand, automated and on-demand update management must ensure that updates are authenticated and applied safely, without violating timing budgets or resource isolation assumptions (ISO 26262). On the other hand, the update pipeline must prevent tampering with, triggering of unauthorized updates, and malicious manipulation of firmware images or metadata (ISO/SAE 21434). The temporaneous disruption incurred during updating and patching must be minimized in order not to break the SASE envelope
Monitoring	May be implemented at SW level for relevant events
Controllability	May need explicit enabling/disabling/configuration in middleware, possibly requiring OS/Hypervisor level support
Metrics	Metrics could be defined on specific mechanisms relying on the property i. Max disruption time ii. Update pending to be applied: <SystemElement, Yes/No> iii. Patch pending to be applied: <SystemElement, Yes/No> iv. Number if applied patches/update v. Date of latest patch/update
Formalization	-
WPs	WP1, WP3, WP4

3.1.26. SASE-Intra-26

Property ID	SASE-Intra-26
Relevance	Security
Related properties	N/A
Scope	Intra-node
Level	SW
Type	Desirable
Description	Software support for a monitoring strategy aiming at intercepting and reacting to suspicious events and patterns. Logs should also be protected from external manipulation
Rationale	Monitoring abnormal events and enforcing protected, tamper-resistant logging is vital for ensuring that security violations and safety issues are promptly intercepted, so that the node remains within its SASE envelope during runtime and OTA updates. While this property is meant to be implemented at the software/services/orchestration layers, it relies on specific support in the hardware node. Besides the monitoring framework, it is fundamental to ensure also the logs generated from monitoring (on violations and issues) monitoring must also be protected from potential external manipulation
Monitoring	May be implemented at SW level for relevant events i. Software-level monitoring: [unsupported, enabled, disabled] ii. Log file status: [safe, compromised]
Controllability	May need explicit enabling/disabling/configuration in middleware, possibly requiring OS/Hypervisor level support
Metrics	i. Number of violations detected ii. Max reaction time (<i>ms</i>) iii. Number of log entries
Formalization	-
WPs	WP1, WP4

3.1.27. SASE-Intra-27

Property ID	SASE-Intra-27
Relevance	Security
Related properties	N/A
Scope	Intra-node
Level	External HW setup
Type	Desirable
Description	System must be protected from physical threats (e.g., access)
Rationale	Physical access to a computing node exposes it to a class of attacks that cannot be mitigated purely through software or logical isolation. Physical attacks are explicitly in scope for ISO/SAE 21434 under tampering, bypass, probing, side-channel extraction, fault injection. The set of mechanisms contributing to this property vary from hardware being designed to detect, resist, or react to physical intrusion (cases, screws, etc.), protection against unauthorized low-level access through debug probes, etc
Monitoring	-

Controllability	-
Metrics	Metrics can be defined on the resistance of mechanisms, reaction time, etc.
Formalization	-
WPs	WP1, WP4

3.1.28. SASE-Intra-28

Property ID	SASE-Intra-28
Relevance	Safety Security
Related properties	N/A
Scope	Intra-node
Level	HW and SW
Type	Mandatory
Description	Hardware and software support to enable the system to react and recover from safety and security failures with minimal disruption of critical services
Rationale	Prompt reaction and recovery is fundamental to ensure that the node can remain inside its SASE operational envelope even in the presence of faults or attacks. Several mechanisms concur to this property, including: fault detection mechanisms, isolation support, safe fallback modes, and controlled recovery strategies that preserve high ASIL functionalities. In case of failure, it is critical to recover the system in the shortest period of time as possible, while causing minimal or no disruption to critical services. From a functional safety perspective. Those mechanisms enable fail-operational or fail-safe transitions and help preventing or containing the impact of timing violations (ISO 26262). From a security perspective, the capability to support prompt attack detection, containment, and secure re-initialization prevent faults from propagating in time and space (ISO/SAE 21434)
Monitoring	For each specific mechanism: i. Mechanism: [unsupported, enabled, disabled]
Controllability	May need explicit enabling/disabling/configuration in middleware, possibly requiring OS/Hypervisor level support
Metrics	Metrics may be defined based on the diversity and redundancy scheme and mechanism. i. Failure Detection Latency ii. Max disruption time iii. Recovery success rate
Formalization	-
WPs	WP1, WP3, WP4

3.1.29. SASE-Intra-29

Property ID	SASE-Intra-29
Relevance	Security
Related properties	SASE-Intra-36
Scope	Intra-node
Level	HW and SW
Type	Desirable

Description	The system (HW and SW) minimizes the attack surface by removing unnecessary components and ensuring the system is up to date.
Rationale	Components, interfaces, services, drivers, libraries, and subsystems that are not required for the operation of critical functions must be systematically removed, deactivated, or isolated to ensure the attack surface exposed by the systems is minimal. Minimization of the attack surface also assumes the system is up to date. Reducing the attack surface decreases the number of ways an attacker can compromise the system.
Monitoring	-
Controllability	May need explicit enabling/disabling/configuration in HW and SW layers, possibly requiring OS/Hypervisor level support.
Metrics	i. Number of disabled elements ii. Residual number of unused elements
Formalization	-
WPs	WP1, WP4

3.1.30. SASE-Intra-30

Property ID	SASE-Intra-30
Relevance	Safety
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware to support a watchdog with separate time-base and time-window.
Rationale	Hardware watchdog is a fundamental safety (and security) mechanism used to maintain the system in a trustworthy state, detect malfunctioning or compromised software, and enforce recovery with minimal disruption of critical services (unresponsive). The implementation of a watchdog is required by IEC 61508-7 and IEC 61508-2, as a safety measure in order to monitor the behavior of the system. The watchdog monitors the system and enforces corrective action if software becomes unresponsive, unstable, or maliciously blocked, whereas the time window (min and max time for the watchdog to be triggered) is crucial to avoid false triggers.
Monitoring	i. Mechanism: [unsupported, armed, triggered]
Controllability	May need explicit enabling/disabling/configuration in middleware, possibly requiring OS/Hypervisor level support
Metrics	i. Recovery Success Rate ii. False positive rate iii. Time-Window Accuracy
Formalization	-
WPs	WP1, WP4

3.1.31. SASE-Intra-31

Property ID	SASE-Intra-31
Relevance	Safety
Related properties	SASE-Intra-33
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support for monitoring the power supply's output
Rationale	As required by IEC 61508-2 and IEC 61508-7 A 8.3, failures in the power supply must be detected in order to take safety measures. For that purpose, hardware support for monitoring the power supply's output must be provided. In the case of a power state deviating away from the nominal range, it must trigger the implemented safety measures
Monitoring	i. Power supply status: [unsupported, undervoltage, overvoltage]
Controllability	-
Metrics	-
Formalization	Yes
WPs	WP1, WP4

3.1.32. SASE-Intra-32

Property ID	SASE-Intra-32
Relevance	Safety
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware shall support a Clock Supervisor capable of detecting anomalies in program execution sequences caused by timing faults or clock malfunctions
Rationale	The implementation of a clock supervisor is required by IEC 61508-2 and serves as detection mechanism to ensure the correct functioning of the system's clock. The deviation between the pre-established clock and the monitored clock can trigger faults in the system if it exceeds the safe operational range
Monitoring	i. Clock frequency status: [unsupported, nominal, deviation]
Controllability	-
Metrics	i. Number of intercepted deviations beyond the operational range
Formalization	Yes
WPs	WP1, WP4

3.1.33. SASE-Intra-33

Property ID	SASE-Intra-33
Relevance	Safety
Related properties	SASE-Intra-31
Scope	Intra-node
Level	HW

Type	Mandatory
Description	Hardware supports power-down with safety shut-off in case of overvoltage or undervoltage so that those scenarios are promptly detected and, when necessary, the internal state can be saved in non-volatile memory
Rationale	Explicitly required by IEC 61508-7 A 8.3. ISO 26262 explicitly requires controlled deactivation strategies for safety-critical E/E systems, to ensure that safety goals are preserved when the system transitions from an operational state into a non-operational or partially deactivated state. A safe power-down must guarantee that all safety-critical tasks and hardware modules move into a defined, predictable, and verified safe state, for example, it must avoid leaving sensors or control units in undefined states or interrupting safety mechanisms mid-execution
Monitoring	i. Voltage state: [unsupported, nominal, over, under]
Controllability	-
Metrics	i. Number of overvoltage detected ii. Number of undervoltage detected
Formalization	Yes
WPs	WP1, WP4

3.1.34. SASE-Intra-34

Property ID	SASE-Intra-34
Relevance	Security
Related properties	N/A
Scope	Intra-node
Level	SW
Type	Desirable
Description	Software support to isolate update execution per partition with enforced Security Levels and communication policies
Rationale	When updating different elements of the system, such as VMs, images or modules, it is important to guarantee the security of the update, so that only trusted code is used in the update. For that purpose, different communication security policies must be enforced, to avoid attackers from tampering with the update. Performing said updates per partition also allows the security measures to focus on a reduced attack surface, making it much harder for attackers
Monitoring	i. Update status per partition: [Pending, Failure, Success]
Controllability	i. <Partition, Update> mapping, requires middleware level support
Metrics	i. Number of attempted updates per partition ii. Number of failed updates per partition iii. Date of last update per partition
Formalization	-
WPs	WP1, WP3, WP4

3.1.35. SASE-Intra-35

Property ID	SASE-Intra-35
Relevance	Security
Related properties	N/A
Scope	Intra-node
Level	SW
Type	Desirable
Description	Software-support to monitor runtime behavior and securely log events for audit and anomaly detection
Rationale	When attackers interfere with the system, either to extract data or to crash the system, traces of their actions affect the system's behavior. It is therefore crucial to actively monitor the system to look out for any unusual behavior, as that might provide indications that there is an intruder in the system. When the system is under stress due to anomalies, it also modifies the behavior of the system. Thus, providing information on where and how an anomaly happens allows tackling the issue efficiently. Storing this information into logs in a secure way is essential so that they are not tampered by attackers trying to delete their traces, allowing them to be audited afterwards
Monitoring	May be implemented at SW level for relevant events i. Software-level monitoring: [unsupported, enabled, disabled] ii. Log file status [safe, compromised]
Controllability	May need explicit enabling/disabling/configuration in middleware, possibly requiring OS/Hypervisor level support
Metrics	i. Number of violations detected ii. Max reaction time (<i>ms</i>) iii. Number of log entries
Formalization	-
WPs	WP1, WP3, WP4

3.1.36. SASE-Intra-36

Property ID	SASE-Intra-36
Relevance	Safety Security
Related properties	SASE-Intra-29
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support for identifying the number and type of active computing elements
Rationale	The information of active computing elements (number and type) is fundamental for safety and security mechanisms and the orchestrator plane to take informed strategic decisions. The system might be composed of a heterogeneous mix of cores, thus, knowing the type of cores in the system, as well as the number of cores per type, is essential for the SASE assessment
Monitoring	For each computing element: i. Component type ii. Component status [enabled, disabled]

Controllability	-
Metrics	i. Number of active computing elements ii. <Computing element, type> mapping
Formalization	Yes
WPs	WP1, WP3, WP4

3.1.37. SASE-Intra-37

Property ID	SASE-Intra-37
Relevance	Safety Security
Related properties	SASE-Intra-39
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support for monitoring the load of different computing elements.
Rationale	Hardware support for monitoring the load of different computing elements (CPU, GPU, NPU, memory controllers, and interconnects) provides the real-time visibility required to detect overload, abnormal behavior, or resource starvation that may compromise safety-critical timing or hint at possible security breaches. From a functional Safety (ISO 26262) perspective, load anomalies can lead to the violation of timing bounds. From a security (ISO/SAE 21434) perspective, load spikes can reveal DoS attacks or malicious computation. Monitoring enables the system to promptly detect execution anomalies early and apply mitigation strategies (task migration, throttling, mode changes, or safe fallback)
Monitoring	For each computing element: i. Load monitoring: [Unsupported, enabled, disabled] ii. Load in monitoring window
Controllability	May need explicit enabling/disabling/configuration in middleware i. <Computing element, max expected utilization> mapping
Metrics	i. Operation per ms, depending on the computing element/device
Formalization	Yes
WPs	WP1, WP3, WP4

3.1.38. SASE-Intra-38

Property ID	SASE-Intra-38
Relevance	Safety Security
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support for tracking the temperature in each computing element.
Rationale	Providing capabilities to capture the temperature of different computing elements improves safety as it allows monitoring whether different computing elements are working in safe temperature ranges. Thus, monitoring the temperature allows taking action on certain components before they experience a fault. Capturing the temperature of each core allows

	knowing whether it is running at dangerous temperatures, as well as it serves as an indication of overloading. If the temperature falls within unusual levels, it could also be an indication that the system is under attack, trying to cause a failure in the computing elements
Monitoring	i. Temperature in each computing element: [Unsupported, Value]
Controllability	-
Metrics	i. Temperature per core in °K ii. Temperature deviation in °K: [inRange, outOfRange]
Formalization	-
WPs	WP1, WP3, WP4

3.1.39. SASE-Intra-39

Property ID	SASE-Intra-39
Relevance	Safety Security
Related properties	SASE-Intra-37
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support for monitoring CPU cache behavior per core and per cache level.
Rationale	Providing hardware support for monitoring the CPU cache behavior is essential to evaluate the performance. As such, there are several metrics to collect in order to acquire a comprehensive overview of the CPU cache behavior. With this collection of metrics, it can be evaluated whether the CPU cache is operating in safe levels, as it is required for it to behave in a predictable way and to react under specific time constraints. For security purposes, monitoring the correct behavior of the CPU cache is crucial to discern whether the cache might be under attack
Monitoring	For each CPU cache level i. Cache statistic per Core
Controllability	i. Configuration register (may require kernel module, kernel capabilities)
Metrics	i. Access per source (core, accelerator...) ii. Accesses per type (load/store) and per source (core, accelerator...) iii. Misses per type (load/store) and per source (core, accelerator...) iv. Latency per type (load/store) and source (core, accelerator...): [Min:Avg:Peak]
Formalization	Yes
WPs	WP1, WP3, WP4

3.1.40. SASE-Intra-40

Property ID	SASE-Intra-40
Relevance	Safety Security
Related properties	SASE-Intra-43
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support for retrieving the different types of memories available in the system and the number of units per type
Rationale	The information of memories (number and type) is fundamental for safety and security mechanisms and the orchestrator plane to take informed strategic decisions. The system might be composed of a heterogeneous mix of memories, thus, knowing the type of memories in the system, as well as the number of memories per type, is essential for the SASE assessment
Monitoring	For each memory element: i. Memory type ii. Memory status [enabled, disabled]
Controllability	-
Metrics	i. Number of memories ii. <Memory, type> mapping
Formalization	Yes
WPs	WP1, WP3, WP4

3.1.41. SASE-Intra-41

Property ID	SASE-Intra-41
Relevance	Safety Security
Related properties	SASE-Intra-43
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support for gathering information of the memory bandwidth utilization
Rationale	The bandwidth utilization of a memory provides essential data of the performance. For safety purposes, it is desired to use the bandwidth in a predictable manner, with higher bandwidth guarantees allocated to higher criticality functions. Uncontrolled bandwidth usage can lead to timing violations, potentially resulting in system malfunctions. From a security perspective, unusual bandwidth utilization levels could be an indicator of an attacker trying to clog a given memory, making it unavailable to other functions
Monitoring	For each memory element: i. Memory bandwidth monitoring: [Unsupported, enabled, disabled] ii. Data processed per memory device
Controllability	May need explicit enabling/disabling/configuration in middleware
Metrics	i. Memory bandwidth utilization as data per ms: [Min:Avg:Peak]
Formalization	Yes
WPs	WP1, WP3, WP4

3.1.42. SASE-Intra-42

Property ID	SASE-Intra-42
Relevance	Safety Security
Related properties	N/A
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support for gathering the number of memory faults experienced by different memories, per type and id
Rationale	Hardware mechanisms that count memory faults, such as ECC correctable errors, uncorrectable errors, parity violations, row hammer events, retention faults, and controller timeout events provide essential information on health and trustworthiness of memories (SRAM, DRAM, flash, NVM, scratchpads, caches) allowing for the detection of possible degradation, anomalous patterns, or malicious activity. For safety reasons, faults can cause malfunctions or make the system behave in an unpredictable manner. For security reasons, the number of faults can be an indicator of DoS attacks, as well as a source of data breaches, allowing attackers to crash the system or to acquire sensitive data that should be protected
Monitoring	For each memory element: i. Memory health monitoring: [Unsupported, Enabled, Disabled] ii. Number of correctable errors iii. Number of uncorrectable errors iv. Number of parity violations v. Number of controller timeouts
Controllability	May need explicit enabling/disabling/configuration with OS/Hypervisor level support i. Configuration register (may require kernel module, kernel capabilities)
Metrics	Metrics can be defined for the specific mechanism supported by counting events over time or relative to memory utilization
Formalization	Yes
WPs	WP1, WP3, WP4

3.1.43. SASE-Intra-43

Property ID	SASE-Intra-43
Relevance	Safety Security
Related properties	SASE-Intra-40, SASE-Intra-41
Scope	Intra-node
Level	HW
Type	Desirable
Description	Hardware support for gather metrics regarding the utilization (percentage) of the storage capabilities of each memory by type and id
Rationale	Hardware mechanisms that report utilization of memory resources (SRAM banks, DRAM channels, scratchpads, caches, NVM blocks, etc.) allow to detect

	abnormal consumption patterns, anticipate overload, identify resource exhaustion, and guarantee that memory requirements of safety-critical and security-critical tasks are met. Monitoring memory utilization is essential for detecting anomalies that could compromise timing determinism, introduce safety-critical timing risks. Overutilization can also hint at memory-exhaustion attacks as overloaded memory can cause delays in information processing, which can hinder critical tasks, compromising the system
Monitoring	For each memory element: i. Memory utilization tracking: [Unsupported,Enabled,Disabled] ii. Memory type iii. Memory status [enabled, disabled] iv. Memory utilization (%)
Controllability	May need explicit enabling/disabling/configuration with OS/Hypervisor level support i. Configuration register (may require kernel module, kernel capabilities)
Metrics	i. Max memory utilization per device (%) ii. Max memory utilization per type (%)
Formalization	-
WPs	WP1, WP3, WP4

3.1.44. SASE-Intra-44

Property ID	SASE-Intra-44
Relevance	Safety Security
Related properties	
Scope	Intra-node
Level	HW and SW
Type	Desirable
Description	The hardware platform should support the replacement or addition of SoM elements or alike while preserving system functionality, safety, security, and performance constraints.
Rationale	The system should also support hardware module changes through reconfigurations and updates, without requiring redesign of the execution platform, and while still providing critical system functionality and preserving safety, security, and performance constraints.
Monitoring	For each new element: i. Element type ii. Element id
Controllability	-
Metrics	-
Formalization	-
WPs	WP1, WP2, WP3, WP4

3.2. Inter-node Level Properties

In the following, we gather a set of node-level properties that pertain to the inter-node (communication) dimension.

3.2.1. SASE-Inter-01

Property ID	SASE-Inter-01
Relevance	Security
Related properties	N/A
Scope	Inter-node
Level	HW and SW
Type	Mandatory
Description	Hardware and Software support to secure in-vehicle and external communication
Rationale	Communication links represent a critical attack surface and a key source of potential safety hazards. Ensuring the authenticity, integrity, confidentiality, and availability of messages across internal buses (CAN, FlexRay, Ethernet) and external interfaces (V2X, OTA, Wi-Fi) is fundamental for maintaining the safe and secure operational envelope of the computing node. Secure communication is fundamental for preventing injection, spoofing, replay, man-in-the-middle attacks. OTA updates fully depend on secure communication as update commands must be authenticated and update payloads must be encrypted and integrity-checked
Monitoring	Depends on specific mechanism
Controllability	May need explicit enabling/disabling/configuration in middleware, possibly requiring OS/Hypervisor level support
Metrics	Metrics can be defined based on the mechanism in place
Formalization	-
WPs	WP1, WP4

3.2.2. SASE-Inter-02

Property ID	SASE-Inter-02
Relevance	Security
Related properties	N/A
Scope	Inter-node
Level	HW and SW
Type	Desirable
Description	Hardware and software support to obtain inter-nodes communication logs. These logs are required to build datasets, which serve as a crucial input for the training of IDS
Rationale	The inter-nodes communication security is fundamental for the system integrity. Inter-node communication logs (e.g., message traces, timestamps, integrity results, and protocol-level events) provide critical information on inter-node functions. Reliable communication logging is required to detect safety-critical timing violations and security attacks. Logs of the communication network are also typically used to train Intrusion Detection Systems (IDS) which are used to monitor the communication network. The storage of the logs of inter-nodes communication must also avoid tampering

Monitoring	i. Inter-node logging: [unsupported, enabled, disabled] ii. Logs status: [safe, compromised]
Controllability	May need explicit enabling/disabling/configuration in middleware, possibly requiring OS/Hypervisor level support
Metrics	i. Number of violations detected ii. Max reaction time (<i>ms</i>) iii. Number of log entries
Formalization	-
WPs	WP1, WP4

3.2.3. SASE-Inter-03

Property ID	SASE-Inter-03
Relevance	Security
Related properties	N/A
Scope	Inter-node
Level	HW and SW
Type	Desirable
Description	Hardware and software support for enabling real-time packet capturing and monitoring of inter-node communications from a specific node for intrusion detection purposes
Rationale	The IDS actively monitors the inter-node communication network in order to detect unusual behavior that serves as indication that an attacker is targeting the system. Thus, the IDS must monitor the inter-node communication network in real-time, capturing the different packets
Monitoring	i. IDS: [unsupported, enabled, disabled] ii. Logs file status: [safe, compromised]
Controllability	May need explicit enabling/disabling/configuration in middleware, possibly requiring OS/Hypervisor level support i. <NodeID,Monitoring Enabled/Disabled> IDS mapping
Metrics	i. Number of violations detected ii. Max reaction time (<i>ms</i>) iii. Number of log entries
Formalization	-
WPs	WP1, WP4

3.2.4. SASE-Inter-04

Property ID	SASE-Inter-04
Relevance	Safety Security
Related properties	N/A
Scope	Inter-node
Level	HW and SW
Type	Desirable

Description	Support for capturing the bandwidth of the network interface
Rationale	The actual bandwidth of the network interface can provide information on the utilization of the network. If the used bandwidth is out of a safe operational range, it could compromise the integrity of the system, as communication delays between different nodes may impair the execution of safety-critical functions. Unusual network bandwidth usage patterns could also serve as an indicator that an attack is occurring in the system.
Monitoring	i. Memory bandwidth monitoring: [Unsupported, enabled, disabled] ii. Data packets sent over the network
Controllability	May need explicit enabling/disabling/configuration in middleware
Metrics	i. Network bandwidth utilization: [Min:Avg:Peak]
Formalization	Yes
WPs	WP1, WP4

3.3.Alignment with other WPs

The initial phases (MS1, MS2) of UP2DATE4SDV are devoted to the identification of a common set of functional and non-functional requirements across all abstraction layers (HAL, OAL, and ORP) in the proposed approach. The requirement gathering process is strongly steered by the specific needs captured by the project use cases and demonstrators addressed in WP5: the first set of requirements identified in the project are those emerging from the use cases as described in D5.1 at the end of month 3 in the project (MS1). Requirements on the abstraction layers, instead, have been collected during the first 6 months and are formalized in the set of MS2 deliverables from WPs 1, 2, and 3.

Within this scenario, D1.1 focuses on the collection of SASE properties from the hardware and low-level software perspectives, whereas D2.1 and D3.1 focus on the requirements defined over the operating and software system layer and on the orchestrator and deployment layer, respectively. As the three abstraction layers are meant to closely cooperate, it is important to guarantee the corresponding set of requirements are not only aligned with the use cases requirements but are also reciprocally consistent.

From WP1 perspective, the relationship between the layers and the requirements stands on the premises that both OAL and the ORP layers require support on the hardware side. On one side, WP1 HAL shall support WP2 SASE-compliant functional mechanisms and WP3 SASE-informed orchestration based on well-defined, observable, and enforceable node-level features and properties. The identified high-level SASE properties contribute to these goals by providing a common taxonomy, clear abstraction boundaries, and measurable observability requirements.

Under the premises that requirements at inter-WP scope are necessarily high level, we performed a cross-validation of requirements as consistency check and to intercept potential misalignment. Our conclusion is that the requirements are consistent with the layered view where HAL is ultimately responsible for providing the observability and controllability knobs to support the upper layers.

4. Summary and Outlook

During the initial phase of WP1 from M1 to M6, an in-person meeting was organized at BSC premises) in Barcelona, where WP partners discussed the main strategy to follow for the identification of platform level properties required to support UP2DATE4SDV goals. Since that, coordination and cooperation among WP partners have been guaranteed through monthly WP meetings.

Deliverable D1.1 gathers the outcomes of Task T1.1 on the Identification of Node-Level Safety and Security-Relevant Properties. T1.1 contributed to the identification of properties at both the node level (e.g., time determinism, partitioning, redundancy) and the inter-node communication level (e.g., data integrity, bounded latency), to enable the definition of metrics and (model-based) representations of these properties, and to establish an early set of requirements for monitoring, diagnostic, and control mechanisms. Ultimately, this initial outcome will steer all WP1 efforts providing the necessary abstractions and APIs for enabling: the hypervisor in WP2 to manage hardware resources, the orchestration layer in WP3 to take decisions based on SASE assessments, and SASE mechanisms in WP4 to ensure safe and secure updates.

The main artifact produced by T1.1. consists in a list of high-level properties capturing the safety and security dimensions in OTA systems updates and reconfiguration. Such list establishes a conceptual taxonomy and defines the set of security- and safety-relevant properties at an abstract level: the feasibility, granularity, and enforceability of these properties are determined by the actual support provided by the underlying platform and will be addressed in other already running tasks in WP1.

In conclusion, D1.1 provides a first identification of hardware level properties that are considered relevant for supporting UP2DATE4SDV goals and enabling hardware and software mechanisms in the upper software layer of the execution stack, to support safe and secure updates and reconfiguration of SDVs in CCAM scenarios.

The next deliverables of WP1 are scheduled in nine month (M15) and one year (M18). D1.2 (SoC observability and controllability channels), due in M15, will provide the detailed list of the observability and controllability channels of the selected SoCs, including qualitative analyses of their values. It will consider the inputs from D.1.1 and identify relevant monitoring and control aspects for the hypervisor, the ORP and the SASE mechanisms, as well as for the integration into the cloud-based development framework. D1.3 (Configurations and monitoring services [Initial]), due in M18, will provide the initial set of SASE-relevant configurations, libraries, and drivers for the SoC to enable TSN communication, resource and network performance monitoring, and to ensure hypervisor-boundaries and security properties. To provide the basics for the definition and implementation of the ORP, a model-based SoC representation will be supported in terms of the initial version of a hardware abstraction layer (HAL).