The Journal of Systems and Software 241 (2026) 112972

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Model-based dependability assessment of fault-tolerant architectures for autonomous driving systems☆

Manuel Drago [a,*], Andrea Bondavalli [a], Paolo Lollini [a], Moritz Antlanger [b], Sascha Drenkelforth [b]

[a] Department of Mathematics and Informatics, University of Florence, Viale Morgagni 65, 50134, Florence, Italy
[b] TTTech Auto AG, Operngasse 17-21, 1040, Vienna, Austria

ARTICLE INFO

Keywords:
Dependability analysis
Fault tolerance
Fail-operational systems
Safety-critical systems
Autonomous driving
Stochastic Activity Networks (SAN)

ABSTRACT

The design of SAE Level 4 autonomous driving systems requires architectures that can tolerate hardware and software faults to maintain dependable operation. To achieve fail-operational behavior, these systems rely on redundancy and fault-tolerance mechanisms that guarantee continuous and resilient operation over time. This study presents a quantitative dependability assessment of four architectural solutions for an SAE Level 4 Highway Pilot: Triple Modular Redundancy, Channel-Wise Doer/Checker/Fallback, Layer-Wise Doer/Checker/Fallback, and AD-EYE. The analysis employs Stochastic Activity Networks (SAN) to model both independent and common-cause failure scenarios, evaluating probabilistic safety and reliability metrics over transient and asymptotic system behavior, along with a temporal indicator represented by the mean time to failure. A comprehensive sensitivity analysis investigates the influence of critical model parameters on these measures to assess the robustness of the results and highlight the main architectural trade-offs. The results provide quantitative evidence of each architecture's strengths and limitations under different operating conditions, supporting the design of dependable and fail-operational autonomous systems.

1. Introduction

Dependability (Avizienis et al., 2004) has become a central concern in the design of autonomous driving systems, whose increasing software complexity and autonomy levels demand safe functionality and service continuity in the presence of faults. At SAE Level 4 (SAE International, 2021), the system must operate without human supervision within a defined operational design domain while ensuring fail-operational capability—i.e., the ability to maintain essential functionality after failures occur.

A variety of architectural approaches have been proposed in both industry and academia to support fail-operational capability in autonomous driving systems, often inspired by fault-tolerance strategies used in aerospace and industrial automation. These can be broadly classified into monolithic, symmetric, and asymmetric designs (Escuela et al., 2023). Monolithic solutions, relying on a single computing channel, are rarely acceptable for Level 4 applications, as they represent a single point of failure and lack inherent fault tolerance. In contrast, symmetric and asymmetric architectures implement redundancy and functional decomposition to achieve fail-operational capabilities, but they exhibit different dependability trade-offs that remain insufficiently quantified in the literature.

To address this gap, this study performs a model-based dependability analysis of representative fault-tolerant architectures for an SAE Level 4 Highway Pilot. A preliminary version of this work appeared in the 25th IEEE International Conference on Software Quality, Reliability and Security[1] (Drago et al., 2025). This article extends the previous study in several directions: it introduces a new asymmetric architecture (AD-EYE); updates the existing architectural models; broadens the sensitivity analysis to include additional critical parameters; incorporates the mean time to failure as a temporal measure; and extends the probabilistic evaluation to both transient and asymptotic system behavior. Each architecture is modeled using Stochastic Activity Networks (SAN) (Meyer et al., 1985; Sanders and Meyer, 2000) in the Möbius framework (Daly et al., 2000; Clark et al., 2001; Deavours et al., 2002) to capture both independent and common-cause failures, and to characterize system evolution towards safe or unsafe states. The objective is to

☆ This article is part of a Special issue entitled: 'Software Dependability' published in The Journal of Systems & Software.

* Corresponding author.
E-mail addresses: manuel.drago@unifi.it (M. Drago), andrea.bondavalli@unifi.it (A. Bondavalli), paolo.lollini@unifi.it (P. Lollini), moritz.antlanger@tttech-auto.com (M. Antlanger), sascha.drenkelforth@tttech-auto.com (S. Drenkelforth).

[1] https://qrs25.techconf.org/.

https://doi.org/10.1016/j.jss.2026.112972
Received 15 November 2025; Received in revised form 1 May 2026; Accepted 26 May 2026
Available online 15 June 2026

provide quantitative insight into how different architectural principles affect the capability of an autonomous driving system to remain safe and operational in the presence of failures.

The remainder of this paper is organized as follows. Section 2 reviews related work on architectural approaches and dependability modeling for autonomous driving systems. Section 3 presents the reference architectures and their corresponding models. Section 4 presents and discusses the quantitative results. Threats to validity are addressed in Section 5, and Section 6 concludes the paper.

2. Related work

Structural redundancy is a well-established strategy to enhance the dependability of safety-critical systems in domains such as aerospace, railway, and industrial automation. In recent years, similar principles have been increasingly adopted in the automotive domain to support fail-operational behavior in autonomous driving systems (Ishigooka et al., 2018; Schmid et al., 2019; Sari, 2020; Stoffel et al., 2024). Several architectural solutions have been proposed – ranging from symmetric redundancy schemes to asymmetric decompositions – with the common goal of maintaining essential functionality in the presence of failures. Given the variety of these solutions, quantitative evaluation is essential to understand the architectural trade-offs and guide system-level design decisions.

A Markov-based comparison of single- and multi-ECU configurations designed for SAE Level 4 automated driving is presented in Julitz et al. (2022), showing how redundancy patterns and component diversity affect overall system reliability under assumptions of independent and common-cause component failures. A Markov reliability model is developed in Kohn (2017) to compare a fail-operational 2oo2 Diagnostic Fail Safe (DFS) steering system with a non-redundant alternative, highlighting the dependability gains achievable through diagnostic diversity and hardware replication. A computer-aided method to synthesize and evaluate hardware redundancy schemes (M-out-of-N) for autonomous driving platforms is proposed in Julitz et al. (2023). The approach combines fault tree modeling with numerical Markov analysis to assess compliance with ISO 26262 (International Organization for Standardization, 2018) reliability targets and to explore trade-offs between redundancy complexity and safety. A model-based approach to the functional safety analysis of autonomous driving system architectures is presented in Kölbl and Leue (2018), enabling the quantitative evaluation of architectural variants and hardware–software mappings using probabilistic fault trees to verify compliance with ISO 26262 safety goals. Similarly, Frigerio et al. (2021) investigates the impact of different automotive architecture topologies, such as domain-based and zone-based solutions, by applying ASIL decomposition techniques to quantitatively evaluate safety-related metrics including failure probability, resource costs, and communication load. In addition, Boubakri and Gamar (2021) proposes redundancy-oriented system architectures for SAE Level 5 vehicles, combining hardware, software, and communication mechanisms to manage ECU failover, thereby highlighting how coordinated redundancy strategies can improve availability and operational safety in connected autonomous vehicles.

These contributions underline the relevance of quantitative evaluation in fault-tolerant automotive design. However, they are typically limited to hardware-level configurations or specific subsystems, and do not examine architectural principles at the system level. In contrast, the present work introduces a comparative dependability study of four full-system architectures using a unified stochastic modeling framework. By covering both failure behavior and model sensitivity, the study aims to support architecture-level decision-making for SAE Level 4 autonomous systems.

3. System architectures and models

This section presents the fault-tolerant system architectures considered in this study and describes how they are modeled for quantitative comparison in an SAE Level 4 Highway Pilot scenario. The analyzed architectures include Triple Modular Redundancy (TMR), Channel-Wise Doer/Checker/Fallback (CDCF), Layer-Wise Doer/Checker/Fallback (LDCF), and the newly introduced AD-EYE architecture. TMR represents a symmetric solution based on redundancy and voting, whereas CDCF, LDCF, and AD-EYE follow asymmetric design principles that emphasize functional decomposition through different mechanisms. These architectures correspond to the reference set identified by the Safety & Architecture working group of The Autonomous (Escuela et al., 2023), which provides a structured selection of representative fail-operational design solutions. In this work, this reference set is used as a common basis to perform a consistent quantitative comparison within a unified modeling framework.

To enable a uniform and comparable analysis across architectures, each system is modeled using Stochastic Activity Networks (SAN) (Meyer et al., 1985; Sanders and Meyer, 2000), a formalism widely adopted for dependability analysis and extending classical Petri Nets. In all models, timed activities are assumed to follow an exponential distribution, implying constant failure rates over time. No restoration or repair mechanisms are included, as the analysis focuses exclusively on the system's degradation behavior and transition towards failure. Each model captures how the state of individual components determines the global state of the system, which can reside in one of three mutually exclusive conditions: working, safe, or unsafe. The working state corresponds to nominal operation, the safe state represents a controlled system stop, and the unsafe state denotes a loss of control or hazardous outcome. Transitions among these conditions are driven by the occurrence of both independent and common-cause failures of system component, which are explicitly represented in the models. Common-cause failures are assumed to affect only complex components,[2] which may share software, design, or execution environments, whereas simple components[3] are assumed to be sufficiently isolated and therefore subject only to independent failures. Moreover, for complex components, which may be subject to Byzantine behavior, failures are classified as silent (loss of availability) or erratic (loss of integrity due to incorrect outputs). Simple components, characterized by deterministic and fully verifiable behavior, are instead assumed to fail only silently. These modeling assumptions are intentionally adopted to ensure model tractability and to enable a consistent and controlled comparison focused on architectural design choices. The SAN models are derived from the architectural descriptions provided in the original sources, preserving their main functional components and failure-handling mechanisms while abstracting implementation-specific details.

The following subsections describe each architecture together with its corresponding SAN model, highlighting how architectural design choices are reflected in the modeling abstractions. Complete model definitions are provided as supplementary material.

3.1. Triple Modular Redundancy (TMR)

Triple Modular Redundancy (TMR) is a concrete instance of the N-Modular Redundancy (NMR) pattern (Lyons and Vanderkulk, 1962), in which three parallel subsystems perform the full driving task and submit their outputs to voting logic. The TMR architecture considered in this work is based on the reference model defined in Escuela et al. (2023) (Fig. 1), and includes three complex and two simple subsystems:

[2] Complex subsystems must be assumed to exhibit Byzantine faults, characterized by inconsistent or arbitrary behavior in the presence of failures.

[3] Simple subsystems are fully verifiable – preferably through formal methods – and deterministic in behavior.

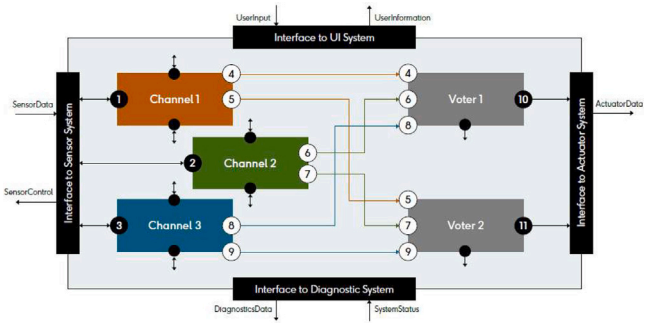

Fig. 1. Block diagram of the TMR architecture.

Table 1
Voters behavior in the TMR architecture.

Channel 1	Channel 2	Channel 3	Voters
A	B~A	C~A	A
A	B~A	C	A
A	B~A	None	A
A	B	C	None
A	B	None	None
A	None	None	A
None	None	None	None

- Channels 1–3: each channel implements the entire autonomous driving processing pipeline—from sensor data acquisition to trajectory planning and actuator setpoint computation. The channels are designed to be functionally equivalent but exhibit software diversity to reduce susceptibility to common-cause failures. This diversity may lead to non-deterministic outputs across channels even under fault-free conditions. To address this, the architecture employs an inexact majority voting strategy.
- Voters 1–2: each voter compares the similarity of the outputs of the three channels and determines the result based on inexact voting. If a majority agreement is found, the selected output is forwarded to the actuator system. Otherwise, the voters remain silent and trigger a pre-buffered Minimal Risk Maneuver (MRM) (SAE International, 2021) (e.g., a controlled deceleration to stop the vehicle). To avoid single points of failure, the voter logic is implemented on two identical, simple, and verifiable units operating in parallel. The complete behavior of the voters is summarized in Table 1, which lists all relevant permutations (for brevity, not all possible permutations are shown; however, the table is designed to be symmetric).

3.1.1. TMR model

The SAN model of the TMR architecture is shown in Fig. 2. The model captures the behavior of the three replicated channels and the two voters described above, focusing on how component failures and output disagreements propagate to the system level.

All components are initially assumed to operate correctly (1 token in each place *Channel1*, *Channel2* and *Channel3*, 2 tokens in the place *Voters*). As time progresses, both channels and voters may fail according to exponential rates. For the replicated channels, the overall failure rate is partitioned into two contributions: a fraction $p_{\text{individual}}$ corresponding to independent failures affecting individual channels (timed activities *Channel1/2/3Failure*), and a remaining fraction corresponding to common-cause failures (timed activity *CCF*) that may simultaneously impact two (output gate *CCF2of3*) or three (output gate *CCF3of3*) channels. When a channel fails, the model distinguishes between different failure modes (instantaneous activities *Channel1/2/3FailureType*). A failed channel may become erratic with probability p_{erratic}, in which case it generates unsafe control commands (place *ErraticChannels*), or silent with the remaining probability, in which case it ceases to produce output (place *SilentChannels*). In the presence of multiple erratic channel failures (input gate *CheckMajorityErratics*), the model distinguishes between cases in which unsafe outputs diverge and cases in which they coincide (instantaneous activity *ErraticResults*). The latter corresponds to a bad majority (place *BadMajority*), i.e., an incorrect decision supported by multiple failed channels, which represents a critical hazard for voting-based architectures. Another aspect explicitly captured by the model is the possibility of output disagreement among correct channels, arising from non-determinism even in fault-free conditions and occurring with rate $r_{\text{disagreement}}$ (timed activity *Disagreement*). Such disagreement may prevent the voting logic from reaching consensus (place *NoMajority*).

A transition to an unsafe state (place *UnsafeState*) occurs when an unsafe decision is accepted, due to a bad majority or because a single erratic channel is the only one still active (input gate *CheckCatastrophicFailure*). Conversely, when no decision can be produced – due to output disagreement, lack of outputs, or voter failures (input gate *CheckNonCatastrophicFailure*) – the execution of a pre-buffered Minimal Risk Maneuver (MRM) potentially brings the system to a safe state (place *SafeState*) with probability p_{MRM}, otherwise leading to an unsafe state.

A breakdown of the TMR failure scenarios is provided in Table 2 (for brevity, not all permutations are shown, but the table is symmetric). All system configurations not explicitly listed correspond to a working system state and are implicitly represented by the model. The complete model documentation is available at https://github.com/Manuel985/AutomotiveArchitecturesSANModels-2-/blob/main/TMR/TMR.pdf.

3.2. Channel-Wise Doer/Checker/Fallback (CDCF)

The Channel-Wise Doer/Checker/Fallback (CDCF) architecture is structured around two architectural design patterns (Koopman, 2020): the Control/Monitor pattern, which separates nominal control from safety monitoring to ensure integrity, and the Duplex pattern, where two control paths operate in parallel to preserve availability, with one designated as primary and the other as backup. The proposed conceptual architecture (Kopetz, 2022) (Fig. 3) consists of three complex and two simple subsystems:

- SAE L2 System: controls the vehicle under nominal conditions. It periodically produces actuator data – such as timed waypoints and corresponding control commands for steering, powertrain, and braking – and sends these outputs to the Monitoring System and to both Decision System instances. If notified that the Fallback Systems is failed, it enters a degraded mode in which only Minimal Risk Maneuvers (MRMs) are produced.
- Monitoring System: receives actuator data from both the SAE L2 and the Fallback Systems. It checks the safety of the received trajectories against its own environment model, and verifies that the actuator data received by the Decision System matches that sent by the control subsystems. This validation process allows the Monitoring System to detect both unsafe behavior and potential Byzantine faults. It sends validation results to the Decision Systems to guide selection.
- Fallback System: operates continuously and generates alternative actuator data aimed at executing a Minimal Risk Maneuver (MRM). Like the SAE L2 System, it transmits this data to the Monitoring System and both Decision Systems. It is only intended to control the vehicle when the SAE L2 System is deemed unsafe.
- Decision System: composed of two identical and simple instances to avoid single points of failure. Each instance receives actuator data from both the SAE L2 System and Fallback System, and echoes it back to the Monitoring System to allow consistency checks. Based on the validation results from the Monitoring System, the Decision System selects which control path to follow and

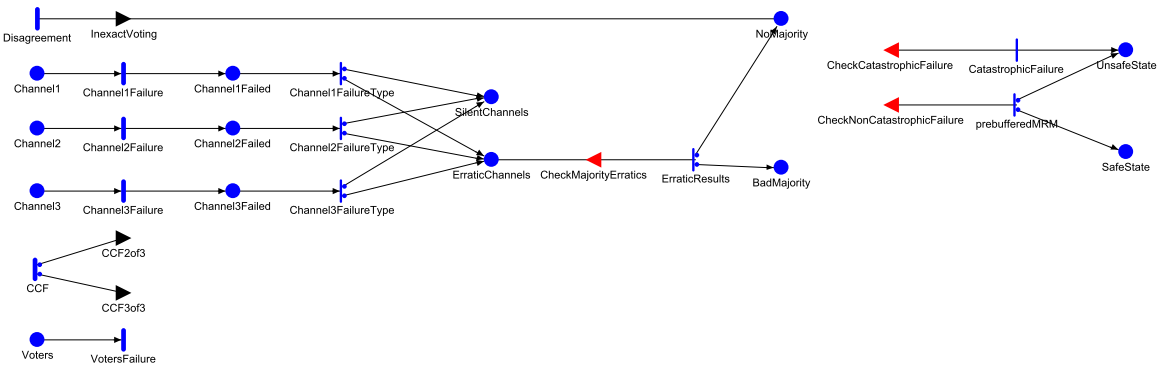

Fig. 2. SAN model of the TMR architecture.

Table 2
TMR failure modes.

Channel 1	Channel 2	Channel 3	#Voters available	Consequence	System state
None	None	None	≥1	Pre-buffered MRM	Safe with probability p_{MRM}
None	A	B	≥1	Pre-buffered MRM	Safe with probability p_{MRM}
A	B	C	≥1	Pre-buffered MRM	Safe with probability p_{MRM}
			0	Pre-buffered MRM	Safe with probability p_{MRM}
None	None	A (unsafe)	≥1	Catastrophic	Unsafe
A (unsafe)	A (unsafe)		≥1	Catastrophic	Unsafe

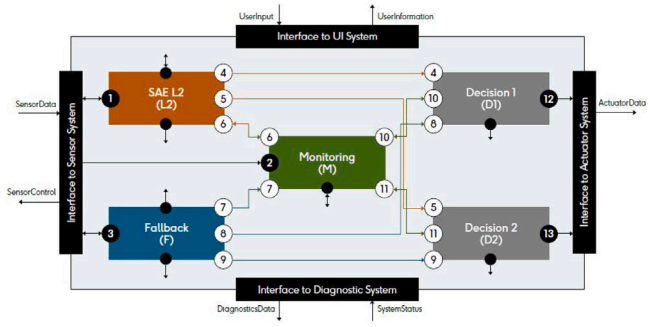

Fig. 3. Block diagram of the CDCF architecture.

forwards the corresponding commands to the Actuator System. By default, the actuator data from the SAE L2 System is selected, but if the Monitoring System detects unsafe behavior, the Decision System switches to the output provided by the Fallback System.

3.2.1. CDCF model

The SAN model of the CDCF architecture is shown in Fig. 4. The model represents the SAE L2, Monitoring and Fallback Systems, and the duplicated Decision modules described earlier, and captures how component failures and incorrect monitoring decisions propagate to the system level.

Initially, all subsystems operate correctly (1 token in each place *SAEL2*, *Fallback*, and *Monitoring*, 2 tokens in the place *Decision*) and the system is in the working state. As time progresses, each subsystem may fail according to exponential rates. For the SAE L2, Monitoring and Fallback Systems, the overall failure rate is partitioned into a fraction $p_{\text{individual}}$ corresponding to independent failures affecting individual components (timed activities *SAEL2/Fallback/MonitoringFailure*), and a remaining fraction accounting for common-cause failures (timed activity *CCF*) that may simultaneously impact two (output gate *CCF2of3*) or three (output gate *CCF3of3*) subsystems. When a failure occurs, the driving units may produce unsafe control commands (places *SAEL2/FallbackErratic*) with probability p_{erratic}, or become silent (places *SAEL2/FallbackSilent*) with the remaining probability. The same failure modeling applies to the Monitoring System, for which erratic behavior corresponds to the issuance of incorrect validations of the SAE L2 System, the Fallback System, or both (instantaneous activity *ValidationError*), leading to false negatives and false positives. A false negative occurs when an unsafe trajectory generated by an erratic driving unit is incorrectly deemed acceptable, potentially leading to the selection of a dangerous control path. Conversely, a false positive results in the rejection of a correct trajectory, which may unnecessarily inhibit a driving unit and trigger degraded behavior.

At the system level, a transition to an unsafe state (place *UnsafeState*) occurs when the vehicle relies on an unsafe trajectory that has been incorrectly validated by the Monitoring System, i.e., when a false negative affects the driving unit controlling the vehicle (input gate *CheckCatastrophicFailure*). Conversely, when the system cannot determine a safe course of action – due to loss of validation feedback, inhibition of both driving units (because they have become silent or have been assessed as unsafe), or failure of both Decision modules (input gate *CheckNonCatastrophicFailure*) – the execution of a pre-buffered Minimal Risk Maneuver (MRM) may bring the system to a safe state (place *SafeState*) with probability p_{MRM}, otherwise leading to an unsafe state. The model also captures degraded but safe operation: when one driving unit is detected as failed (input gates *CheckSAEL2/FallbackMRM*), the remaining module takes over and executes a controlled MRM for a defined period (timed activities *SAEL2/FallbackMRM*), allowing the vehicle to stop safely.

A breakdown of the CDCF failure scenarios is provided in Table 3. All system configurations not explicitly listed correspond to a working system state and are implicitly represented by the model. The complete model documentation is available at https://github.com/Manuel985/AutomotiveArchitecturesSANModels-2-/blob/main/CDCF/CDCF.pdf.

3.3. Layer-Wise Doer/Checker/Fallback (LDCF)

The Layer-Wise Doer/Checker/Fallback (LDCF) architecture follows a hierarchical decomposition based on stacked Control/Monitor pairs (Koopman, 2020), arranged in distinct layers. Each layer integrates a functional unit and a safety monitor, allowing operation to gracefully degrade in case of failures. Specifically, the architecture includes a Primary Layer, which governs nominal operation, and a Safing

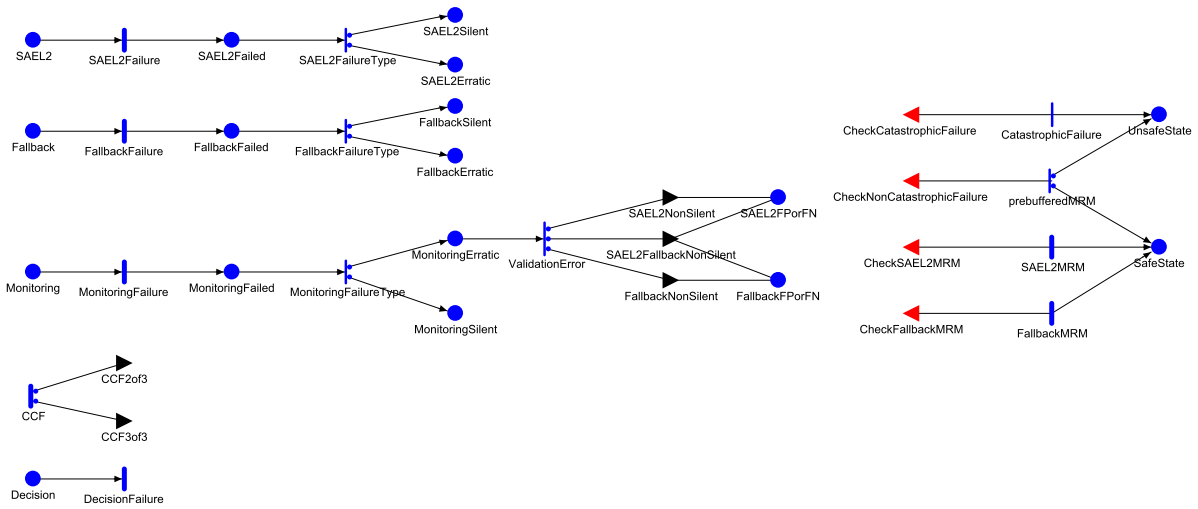

Fig. 4. SAN model of the CDCF architecture.

Table 3
CDCF failure modes.

SAE L2 system	Monitoring system	Fallback system	Decision system	Consequence	System state
	None		Available	Pre-buffered MRM	Safe with probability p_{MRM}
None/Deemed unsafe		None/Deemed unsafe	Available	Pre-buffered MRM	Safe with probability p_{MRM}
			Unavailable	Pre-buffered MRM	Safe with probability p_{MRM}
None/Deemed unsafe	B accepted	B (safe)	Available	Fallback MRM	Safe
A (safe)	A accepted	None/Deemed unsafe	Available	SAE L2 MRM	Safe
None/Deemed unsafe	B accepted	B (unsafe)	Available	Catastrophic	Unsafe
A (unsafe)	A accepted		Available	Catastrophic	Unsafe

Layer, which ensures minimal risk maneuvers (MRMs) if the Primary Layer fails. The proposed conceptual architecture (Wagner et al., 2017) (Fig. 5) consists of four complex and two simple subsystems:

- Primary System: controls the vehicle under nominal conditions. It periodically generates trajectories and actuator setpoints and transmits these to the Primary Safety Gate.
- Primary Safety Gate: complements the Primary System to form a Control/Monitor pair. It checks the outputs of the Primary System and sends validation results to the Priority Selectors.
- Safing System: controls the vehicle when the Primary System is deemed unsafe, ensuring the vehicle reaches a safe state by executing a Minimal Risk Maneuver (MRM). Like the Primary System, it periodically generates trajectories and actuator setpoints, transmitting them to the Safing Safety Gate.
- Safing Safety Gate: complements the Safing System to form another Control/Monitor pair. It checks the outputs of the Safing System and sends validation results to the Priority Selectors.
- Priority Selectors 1–2: decide which setpoints to forward to the vehicle's Actuator System, based on the verdict of the Safety Gates. The Primary System output is the default, but they switch to the Safing System output when the Primary Safety Gate detects an unsafe trajectory generated by the Primary System. To ensure availability, this module consists of two identical instances working in parallel.

3.3.1. LDCF model

The SAN model of the Layer-Wise Doer/Checker/Fallback (LDCF) architecture is shown in Fig. 6. The model represents the Primary and Safing Layers described above, including the corresponding controllers, safety gates, and Priority Selectors, and captures how component failures and incorrect validation decisions propagate to the system level.

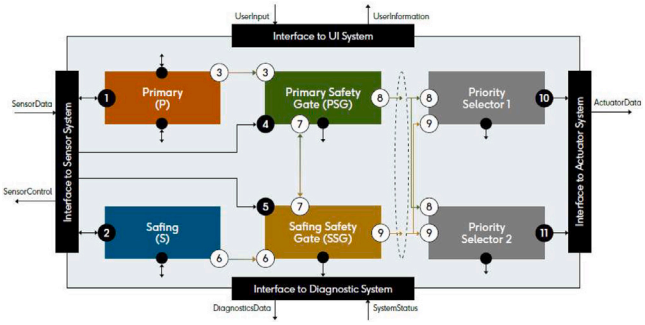

Fig. 5. Block diagram of the LDCF architecture.

At the beginning, all subsystems are considered to function properly (1 token in each place *Primary*, *PrimarySG*, *Safing* and *SafingSG*, 2 tokens in the place *PrioritySelectors*) and the system is in working state. Over time, each subsystem may experience a failure according to exponential rates. For controllers and safety gates, the overall failure rate is partitioned into a fraction $p_{\text{individual}}$ corresponding to independent failures affecting individual components (timed activities *Primary/PrimarySG/Safing/SafingSGFailure*), and a remaining fraction accounting for common-cause failures (timed activity *CCF*) that may simultaneously impact two (output gate *CCF2of4*), three (output gate *CCF3of4*) or four (output gate *CCF4of4*) subsystems. When the Primary or Safing System fails, it may become erratic (place *Primary/SafingErratic*) with probability p_{erratic}, or silent (place *Primary/SafingSilent*) with the remaining probability. The same logic applies to Primary and Safing Safety Gates, where erratic failures can cause false negatives or false positives of the corresponding controller. A false negative occurs when an unsafe trajectory is incorrectly validated as safe, allowing dangerous commands to pass through. Conversely, a false positive leads

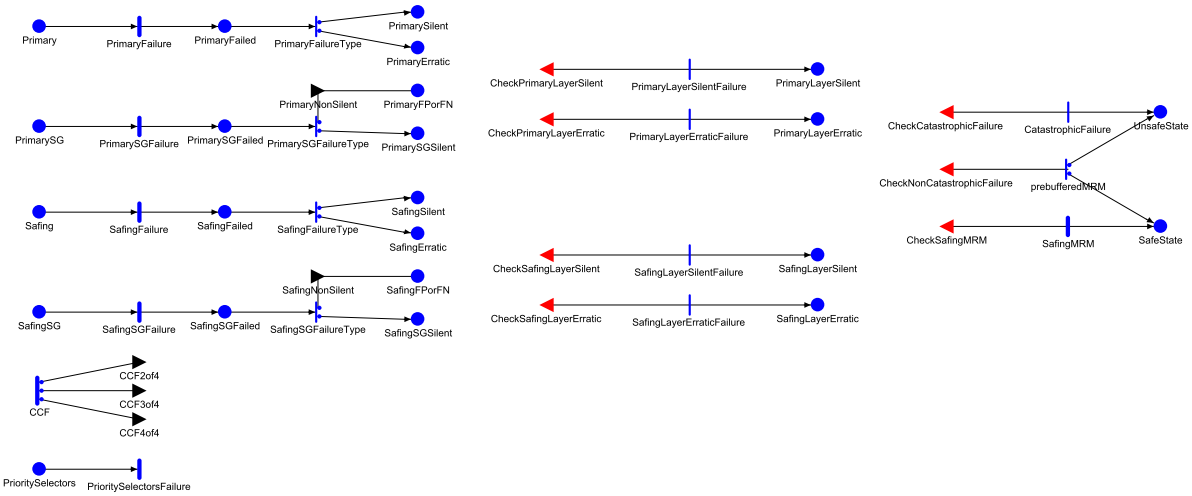

Fig. 6. SAN model of the LDCF architecture.

Table 4
LDCF failure modes.

Primary layer	Safing layer	# Priority selectors available	Consequence	System state
None/Deemed unsafe	None/Deemed unsafe	≥1	Pre-buffered MRM	Safe with probability p_{MRM}
		0	Pre-buffered MRM	Safe with probability p_{MRM}
None/Deemed unsafe	B (safe) accepted	≥1	Safing MRM	Safe
A (unsafe) accepted		≥1	Catastrophic	Unsafe
None/Deemed unsafe	B (unsafe) accepted	≥1	Catastrophic	Unsafe

to the rejection of a valid trajectory, potentially triggering unnecessary degraded behavior.

A catastrophic failure is triggered when a false negative affects the layer controlling the vehicle (input gate *CheckCatastrophicFailure*), resulting in an unsafe trajectory being selected. In such cases, the vehicle transitions to an unsafe state (place *UnsafeState*). If the system cannot determine a valid control path – due to both layers being inhibited (because their controllers or monitors have become silent, or the controllers have been assessed as unsafe), or due to the failure of both Priority Selectors (input gate *CheckNonCatastrophicFailure*) – a pre-buffered Minimal Risk Maneuver (MRM) is executed in an attempt to bring the vehicle to a safe state (place *SafeState*) with probability p_{MRM}, otherwise leading to an unsafe state. When the Primary Layer becomes inhibited (input gate *CheckSafingMRM*), the Safing Layer assumes control and executes a controlled MRM for a predefined time window (timed activity *SafingMRM*) to bring the vehicle to a safe stop.

A breakdown of the LDCF failure scenarios is provided in Table 4. All system configurations not explicitly listed correspond to a working system state and are implicitly represented by the model. The complete model documentation is available at https://github.com/Manuel985/AutomotiveArchitecturesSANModels-2-/blob/main/LDCF/LDCF.pdf.

3.4. AD-EYE

The AD-EYE architecture is another variant of the Control/Monitor combined with Duplex pattern (Koopman, 2020). It adopts an asymmetric structure composed of two independent channels that assume distinct functional roles: nominal control, behavioral supervision through constraint enforcement, and fallback execution. To reinforce separation of concerns, the supervisory and fallback responsibilities are structurally decoupled within one of the channels. The proposed conceptual architecture (Mohan and Törngren, 2019) (Fig. 7) consists of three complex and two simple subsystems:

- Channel 1: controls the vehicle under nominal conditions. It periodically produces actuator data and sends these outputs to the Selectors. If notified that Channel 2 is failed, it enters a degraded mode in which only Minimal Risk Maneuvers (MRMs) are produced.
- Channel 2: includes two structurally independent components that fulfill supervisory and fallback functions.
  - Channel 2a: monitors the behavior of Channel 1 by evaluating environmental conditions and platform diagnostics through a simplified perception stack. It defines and continuously updates a permissive envelope—a set of dynamic constraints that bound the operating conditions under which Channel 1 is allowed to function. If Channel 1 does not meet these constraints, Channel 2a instructs the system to transfer control to Channel 2b.
  - Channel 2b: provides fallback functionality when Channel 1 is considered no longer safe or operational. It generates Minimal Risk Maneuver (MRM) trajectories based on prevalidated paths and simplified perception.
- Selectors 1–2: decide which actuator data is forwarded to the Actuator System. The output of Channel 1 is used by default; however, if Channel 2a detects a violation of the permissive envelope, the Selector switches to the output from Channel 2b. The module consists of two identical instances working in parallel to ensure fault tolerance and continuous operation.

3.4.1. AD-EYE model

The SAN model of the AD-EYE architecture is shown in Fig. 8. The model represents the two-channel structure described above, including Channel 1, the supervisory and fallback units of Channel 2, and the Selector modules, and captures how component failures and incorrect supervisory decisions propagate to the system level.

Initially, all components operate correctly (1 token in each place *Channel1*, *Channel2a* and *Channel2b*, 2 tokens in the place *Selectors*), so the system is in the working state. Over time, each subsystem may experience a failure according to exponential rates. For the channels,

Table 5
AD-EYE failure modes.

Channel 1	Channel 2a	Channel 2b	# Priority selectors available	Consequence	System state
	None		≥1	Pre-buffered MRM	Safe with probability p_{MRM}
None/Envelope violation		None	≥1	Pre-buffered MRM	Safe with probability p_{MRM}
			0	Pre-buffered MRM	Safe with probability p_{MRM}
A (safe)	A meets the envelope	None	≥1	Channel 1 MRM	Safe
None/Envelope violation		B (safe)	≥1	Channel 2b MRM	Safe
A (unsafe)	A meets the envelope		≥1	Catastrophic	Unsafe
None/Envelope violation		B (unsafe)	≥1	Catastrophic	Unsafe

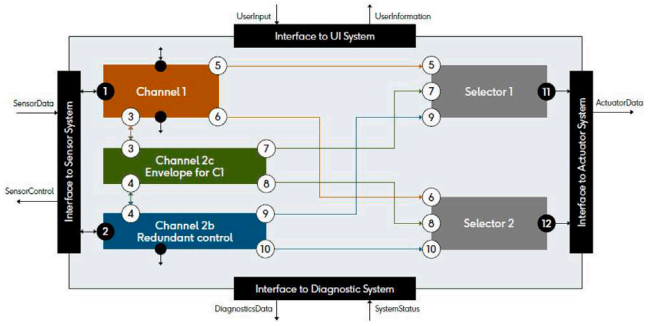

Fig. 7. Block diagram of the AD-EYE architecture.

the overall failure rate is partitioned into a fraction $p_{\text{individual}}$ corresponding to independent failures affecting individual channels (timed activities *Channel1/2a/2bFailure*), and a remaining fraction accounting for common-cause failures (timed activity *CCF*) that may simultaneously impact two (output gate *CCF2of3*) or three (output gate *CCF3of3*) channels. Channel 1 and Channel 2b can exhibit erratic behavior by producing unsafe trajectories with probability p_{erratic}, or fail silently with the remaining probability. Channel 2a follows the same failure modeling approach, where erratic behavior can introduce two potential hazards: a permissive envelope that is too restrictive, which may wrongly inhibit a safe trajectory from Channel 1, and an envelope that is too permissive, which may allow unsafe behavior to go unchecked.

A catastrophic failure occurs in two situations: when Channel 1 produces an unsafe trajectory that is incorrectly accepted due to an overly permissive envelope from Channel 2a; or when Channel 1 is inhibited (either because it has become silent or has violated the envelope) and Channel 2b, though active, generates an unsafe output (input gate *CheckCatastrophicFailure*). In both cases, the Actuator System executes a hazardous command, driving the vehicle into an unsafe state (place *UnsafeState*). A non catastrophic failure is triggered when no eligible driving unit is available, there is a loss of permissive envelope, or because both Selectors have failed (input gate *CheckNonCatastrophicFailure*). In these scenarios, the system falls back to executing a pre-buffered Minimal Risk Maneuver (MRM) in an attempt to reach a safe state (place *SafeState*) with probability p_{MRM}, otherwise leading to an unsafe state. The model also supports controlled degradation. If one of the driving units is notified as failed (input gates *CheckChannel1/2bMRM*), the remaining driving unit takes over and performs a controlled MRM for a defined time window (timed activities Channel1/2bMRM), ensuring the vehicle reaches a safe stop.

A breakdown of AD-EYE failure scenarios is provided in Table 5. All system configurations not explicitly listed correspond to a working system state and are implicitly represented by the model. The complete model documentation is available at https://github.com/Manuel985/AutomotiveArchitecturesSANModels-2-/blob/main/ADEYE/ADEYE.pdf.

4. Results & discussion

This section presents the results of the comparative assessment of four fault-tolerant architectures – Triple Modular Redundancy (TMR), Channel-Wise Doer/Checker/Fallback (CDCF), Layer-Wise Doer/Checker/Fallback (LDCF), and AD-EYE – modeled for an SAE Level 4 Highway Pilot scenario. All results and observations discussed in this section are therefore specific to the considered operational design domain and associated assumptions.

We first introduce the dependability metrics adopted for the evaluation, followed by a description of the evaluation setup. The analysis builds on the SAN models introduced earlier and implemented in Möbius (Daly et al., 2000; Clark et al., 2001; Deavours et al., 2002), and concludes with a discussion of the main results and the architectural trade-offs that emerge from the comparison.

4.1. Metrics of interest

To ensure a consistent and meaningful comparison across the architectures, we adopt a unified set of dependability metrics applicable to all models. These metrics capture both the system's transient and asymptotic behavior:

- $P_{UnsafeState}(t)$: Probability of reaching the unsafe state within the interval $[0,t]$. It is obtained through an instant-of-time reward variable that returns 1 upon entering the absorbing unsafe state, and represents the likelihood of catastrophic failure over time.
- $P_{SafeState}(t)$: Probability of reaching the safe state within the interval $[0,t]$. It is obtained through an instant-of-time reward variable that returns 1 upon entering the absorbing safe state, and represents the likelihood of non-catastrophic termination over time.
- $S(t)$: Probability that the system has not reached the unsafe state in the interval $[0,t]$. This reflects the notion of safety, i.e., absence of catastrophic consequences (Avizienis et al., 2004), and is derived as:

$$S(t) = 1 - P_{UnsafeState}(t)$$

- $R(t)$: Probability that the system has remained operational within $[0,t]$. This corresponds to reliability,[4] i.e., continuity of correct service (Avizienis et al., 2004), and is derived as:

$$R(t) = 1 - P_{UnsafeState}(t) - P_{SafeState}(t)$$

In addition to probabilistic metrics, we evaluate a temporal indicator that characterizes the time dynamics of system degradation:

- $MTTF$ (Mean Time To Failure): Expected time for the system to lose its operational capability, indicated as:

$$MTTF = \mathbb{E}[TTF_{SafeOrUnsafe}]$$

where $TTF_{SafeOrUnsafe}$ denotes the time to absorption in either the safe or unsafe state. This metric provides a compact summary of the expected operational lifetime of each architecture under the considered

[4] In the absence of repairs, reliability also coincides with availability (Avizienis et al., 2004).

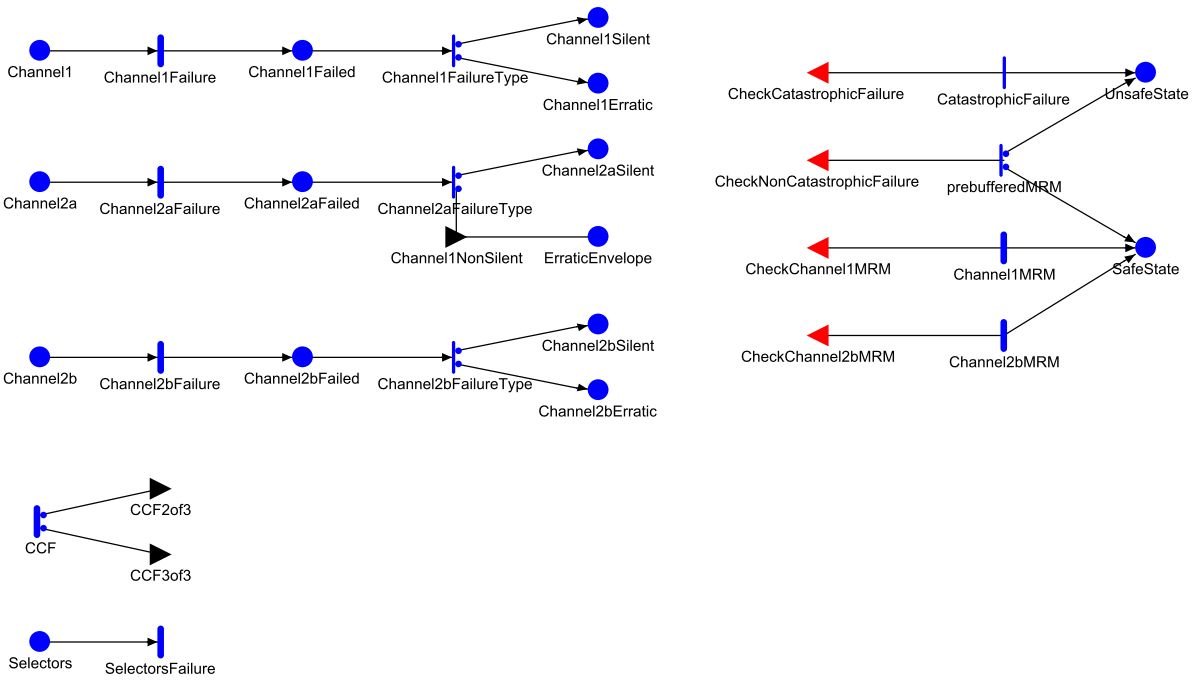

Fig. 8. SAN model of the AD-EYE architecture.

failure assumptions and is computed using standard techniques for mean time to absorption in stochastic models (Trivedi, 2016).

All probabilistic and temporal dependability metrics were computed using the analytic solvers provided by the Möbius modeling framework. Specifically, the Transient Solver was used for transient probabilistic metrics, the Takahashi Steady-State Solver for asymptotic probabilistic metrics, and the Iterative Steady-State Solver for the temporal metric. The solver configuration parameters were set to the default values recommended by Möbius, as provided by the tool, and were consistently applied across all evaluated architectures to ensure comparability of the results. Additional details on the numerical solvers available in Möbius are reported in https://www.mobius.illinois.edu/wiki/index.php/Solving_Models#Numerical_Solvers.

4.2. Evaluation setup

To enable a fair and meaningful comparison across the four fault-tolerant architectures, the SAN models were configured with a shared set of parameters, ensuring that observed differences in dependability metrics stem from the architectural design rather than inconsistencies in modeling assumptions. When needed, additional architecture-specific parameters were included to capture relevant behavioral nuances.

All architectures were modeled using the same failure rates for their constituent components. Complex subsystems, responsible for functions such as trajectory generation, monitoring, and fallback execution, were assigned a failure rate of 10^{-5} per hour. This rate accounts for both independent and common-cause failures. Simple subsystems, including voters and selectors, were modeled with a lower failure rate of 10^{-6} per hour. Moreover, simple subsystems do not directly influence the system's decision-making logic; when a failure occurs, they simply initiate execution of a pre-buffered Minimal Risk Maneuver (MRM). Since this behavior is consistent across all architectures, their failure rates were held constant. The use of identical failure rates across architectures reflects the absence of established reference data for assigning architecture-specific reliability values to such subsystems. In real systems, components in different architectures may exhibit different failure rates due to implementation and design choices. However, without reliable data to support such distinctions, introducing differentiated values would require arbitrary assumptions and could bias the comparison. For this reason, homogeneous failure rates were adopted to ensure a fair and controlled evaluation of architectural design effects.

Table 6
Parameters and value sets used in the sensitivity analysis.

Parameter	Applies to	Values
$p_{\text{individual}}$	All architectures	{0.80, 0.875, 0.95}
p_{erratic}	All architectures	{0.10, 0.30, 0.50}
p_{MRM}	All architectures	{0.75, 0.85, 0.95}
$r_{\text{disagreement}}$	TMR only	$\{10^{-4}, 10^{-5}, 10^{-6}\}$

Building on this common configuration, a systematic sensitivity analysis was performed to evaluate how variations in key modeling parameters affect the performance and robustness of the proposed architectures. The selected parameters represent the main sources of uncertainty and variability in the system models—factors expected to have the strongest influence on dependability over time, and include:

- *p_individual* (all architectures): Fraction of complex component failures that occur independently, with the remaining portion corresponding to common-cause failures.
- *p_erratic* (all architectures): Probability that a failed complex component generates incorrect outputs instead of failing silently.
- *p_MRM* (all architectures): Probability that the pre-buffered Minimal Risk Maneuver (MRM) successfully brings the vehicle to a safe stop.
- *r_disagreement* (TMR only): Rate of output divergence among non-deterministic TMR channels in nominal conditions.

Table 6 summarizes the values adopted for each parameter. This configuration space provides a coherent basis for comparing the architectures and analyzing their performance trends, robustness, and design trade-offs.

System behavior was evaluated at five representative time points – 5000, 15,000, 25,000, and 35,000 operating hours – as well as in an asymptotic regime. These horizons span short-, medium-, and long-term operating conditions, providing a comprehensive view of the

Fig. 9. Safety of the four architectures under the baseline configuration.

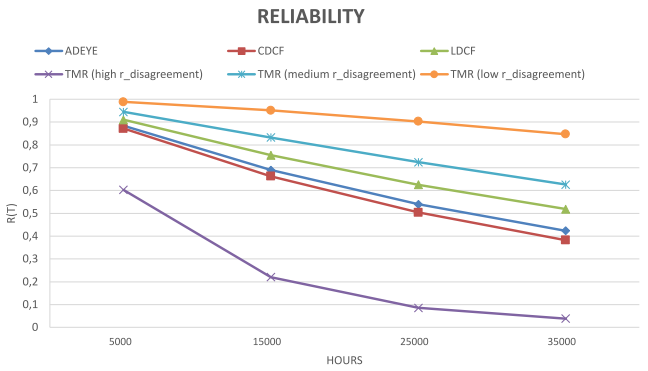

Fig. 10. Reliability of the four architectures under the baseline configuration.

architectures' dependability evolution, from initial operation to their long-term behavior.

4.3. Baseline analysis

This subsection establishes a baseline comparison among the four architectures – TMR, CDCF, LDCF, and AD-EYE – by fixing all common configurable parameters to representative values. The objective is to provide a consistent reference point for assessing their dependability behavior before introducing parameter sensitivity in the following analyses. Specifically, the proportion of independent component failures, i.e., $p_{\text{individual}}$, was set to 0.875; the probability of erratic component behavior upon failure, i.e., p_{erratic}, to 0.3; and the success probability of the pre-buffered MRM, i.e., p_{MRM}, to 0.85. For the TMR architecture, which additionally models disagreement among correct channels due to non-determinism, i.e., $r_{\text{disagreement}}$, we considered three distinct per-hour failure rates—high (10^{-4}), medium (10^{-5}), and low (10^{-6}).

The safety analysis (Fig. 9) reports the safety evolution of all considered architectures over the selected mission time horizon. LDCF achieves the highest safety overall, followed by CDCF, which shows a stable behavior and becomes superior to TMR with low disagreement as mission time progresses. The other architectures follow with lower safety values: AD-EYE, TMR with medium disagreement, and TMR with high disagreement, which attains the lowest safety. A key outcome is that LDCF and TMR with low and medium disagreement exhibit a more pronounced decline over time, reflecting their tendency to prioritize availability by prolonging operation under degraded conditions. In TMR, continued operation with reduced redundancy increases the probability of unsafe decisions, e.g., due to a bad majority or to the erroneous behavior of the last remaining channel. Similarly, in LDCF, failures in the safing layer may still allow operation through the primary layer, but subsequent failures can lead to unsafe conditions, explaining the visible reduction in safety. Conversely, CDCF shows a more constant trend, since any single failure triggers an emergency action, limiting exposure to degraded states. AD-EYE exhibits an intermediate trajectory, consistent with a less aggressive failure-handling policy than CDCF. Finally, TMR with high disagreement yields the lowest safety and shows a step-wise decrease, due to the frequent activation of the pre-buffered MRM induced by its difficulty in forming a majority. Since the pre-buffered MRM may fail, repeated emergency maneuvers can occasionally lead to unsafe states, causing safety drops. Over time, the curve stabilizes as the vehicle progressively loses availability and remains stopped, limiting further exposure to unsafe decisions.

The reliability results (Fig. 10) complement the safety analysis by highlighting the different levels of operational continuity across architectures. TMR configurations with low and medium disagreement clearly dominate, achieving the highest reliability throughout the considered horizon. This behavior is expected, as TMR is explicitly designed to ensure operational continuity under partial degradation by exploiting redundancy and majority voting. LDCF follows with the next highest reliability, for similar reasons: its layered design enables continued operation as long as the primary layer remains functional, allowing the vehicle to tolerate a subset of failures without immediately stopping. AD-EYE achieves intermediate reliability and remains superior to CDCF, which exhibits lower values due to its more aggressive failure-handling policy, which prioritize conservative behavior at the expense of operational continuity. Finally, TMR with high disagreement attains the lowest reliability, as frequent interruptions lead to early loss of operability, thus significantly reducing availability.

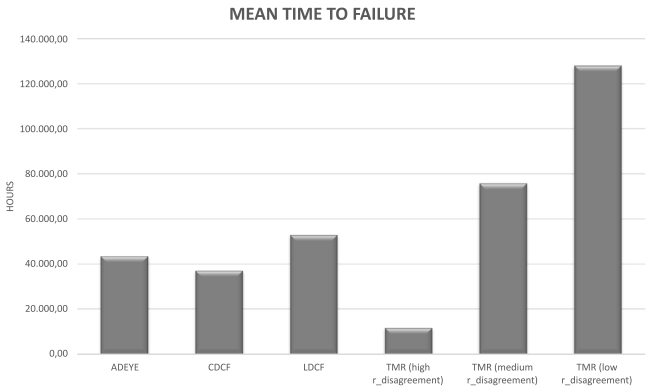

Fig. 11. Mean time to failure of the four architectures under the baseline configuration.

The mean time to failure (Fig. 11) provides an additional perspective on the operational longevity of the considered architectures. The TMR variants with low and medium disagreement achieve the highest MTTF, confirming the benefit of redundancy and voting-based continuity. LDCF follows with intermediate performance, thanks to its layered design. Conversely, AD-EYE and CDCF exhibit shorter mission durations, consistent with more conservative failure-handling strategies. Finally, TMR with high disagreement attains the lowest MTTF, indicating that this configuration is more prone to early termination and thus provides the shortest operational lifetime among the analyzed solutions.

The asymptotic behavior analysis (Fig. 12) compares the long-term termination outcome of the considered architectures, reporting the probability of converging to a safe versus unsafe terminal state as $t \to \infty$. The results show that AD-EYE achieves the highest probability of safe termination, followed by CDCF, indicating that these two architectures are the most likely to eventually end the mission in a safe condition. The fact that AD-EYE slightly outperforms CDCF suggests that, despite CDCF's conservative and aggressive failure-handling strategy, frequent emergency transitions may expose the system more often to termination-related unsafe outcomes, slightly increasing the unsafe terminal probability with respect to AD-EYE. Among the TMR variants, TMR with high disagreement exhibits a higher probability

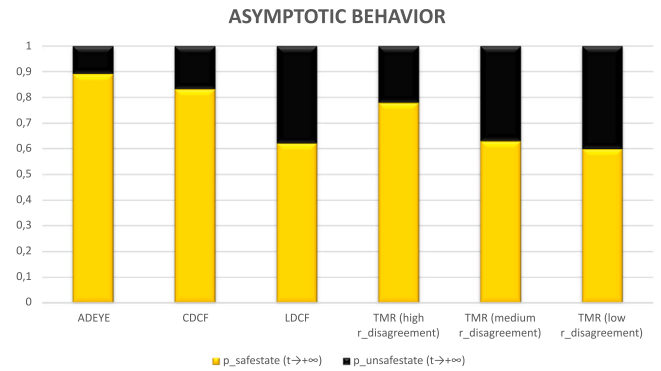

Fig. 12. Asymptotic behavior of the four architectures under the baseline configuration.

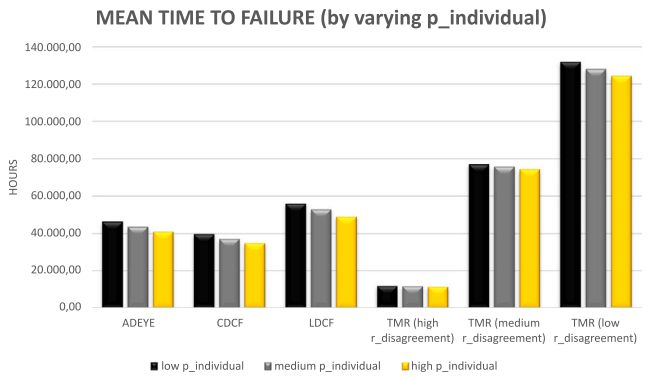

Fig. 13. Mean time to failure of the four architectures by varying $p_{\text{individual}}$.

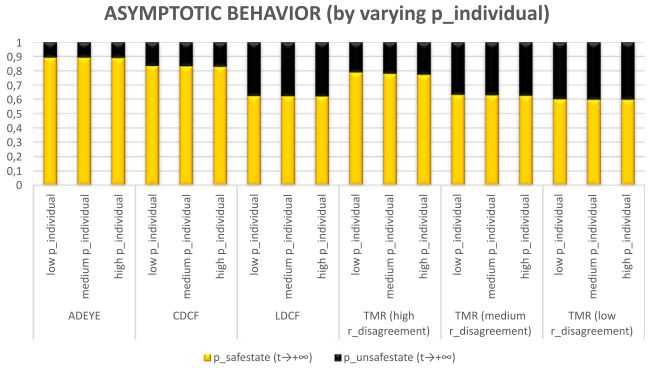

Fig. 14. Asymptotic behavior of the four architectures by varying $p_{\text{individual}}$.

of safe termination than the medium- and low-disagreement cases, whereas TMR with low and medium disagreement show larger unsafe terminations. This behavior is consistent with the fact that, when disagreement is limited, the TMR system tends to remain operational for longer under degradation, increasing long-term exposure to unsafe terminal outcomes. Conversely, higher disagreement leads to more frequent interruptions, which reduces prolonged operation and shifts the asymptotic outcome towards safe termination. Finally, LDCF exhibits an asymptotic behavior comparable to TMR with low disagreement. This suggests that architectures enabling continued operation under partial degradation – while beneficial for operational continuity – may increase the probability of unsafe termination over very long horizons.

Taken together, the transient and asymptotic probabilistic indicators, together with the temporal metric, provide a complete baseline characterization of the dependability of all architectures under the selected parameter settings within the considered SAE Level 4 Highway Pilot scenario. This baseline will serve as the reference point for the upcoming sensitivity analyses, in which the impact of varying key parameters will be assessed relative to the behaviors observed here. For readability, the transient evolution is not reported in the main text for the sensitivity analyses, as comparing multiple architectures across multiple parameter values would require an excessive number of curves and figures. The discussion in the main text is therefore limited to aggregated indicators (MTTF and asymptotic termination probabilities), while the corresponding transient results are reported in Appendix.

4.4. Effect of the independent failure proportion

This subsection analyzes how varying the fraction of independent component failures ($p_{\text{individual}}$) affects the dependability behavior of the four architectures. Three representative values – low (0.85), medium (0.875), and high (0.95) – were considered to capture the sensitivity to the balance between common-cause and independent failure mechanisms.

In terms of mean time to failure (Fig. 13), all architectures show a clear decreasing trend: as $p_{\text{individual}}$ increases, the MTTF systematically decreases. Although this behavior may appear counterintuitive at first, it can be explained by observing that lowering $p_{\text{individual}}$ increases the fraction of common-cause failures, but such failures still represent a relatively smaller component of the overall failure process and tend to occur later over time. In the adopted failure model, the rate of individual failures scales as $\lambda \times p_{\text{individual}}$, whereas the rate of common-cause failures scales as $\lambda \times (1 - p_{\text{individual}})$. Since $p_{\text{individual}}$ is typically high, increasing it significantly raises the effective rate of independent component failures, causing earlier degradation and therefore reducing the mean time to failure. Conversely, when $p_{\text{individual}}$ is lower, the increase in common-cause failures does not compensate for the reduction in independent failures, resulting in a longer operational lifetime and higher MTTF. Importantly, while the absolute MTTF values change with $p_{\text{individual}}$, the relative ordering among architectures remains unchanged with respect to the baseline MTTF results. This indicates that varying the balance between individual and common-cause failures mainly affects the overall failure timescale, without altering the intrinsic robustness ranking of the considered architectures. A further insight emerges when focusing on the three TMR variants. The high-disagreement TMR shows an almost negligible variation, and the medium-disagreement TMR exhibits a much milder decrease than the other architectures. This suggests that, for these configurations, mission termination is often driven by disagreement-related events, which may occur even in the absence of component failures, making MTTF less dependent on the balance between individual and common-cause failures.

The asymptotic behavior analysis (Fig. 14) indicate that the asymptotic safe/unsafe termination split remains essentially unchanged across low, medium, and high $p_{\text{individual}}$ for all architectures. This confirms that the sensitivity on this parameter does not affect the failure-handling logic of the architectures, but mainly rescales the time to failure, i.e., it accelerates or delays the occurrence of failures without altering the long-term termination outcome.

4.5. Effect of erratic behavior probability

This subsection investigates how the probability of erratic behavior upon failure (p_{erratic}) influences the dependability behavior of the four architectures. Three representative values – low (0.1), medium (0.3), and high (0.5) – were considered to capture different proportions of erratic versus silent failure modes.

The mean time to failure results (Fig. 15) show different trends across all architectures. A notable outcome is that AD-EYE and CDCF surprisingly exhibit an increase in MTTF as p_{erratic} grows. This behavior can be explained by the different emergency responses triggered by

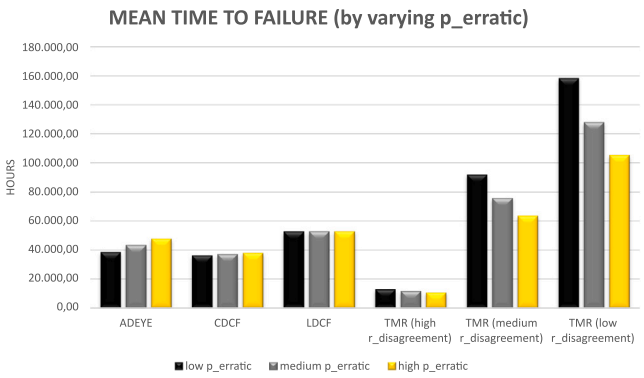

Fig. 15. Mean time to failure of the four architectures by varying p_{erratic}.

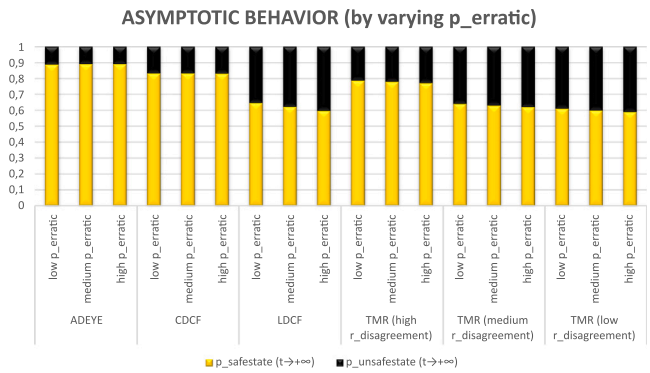

Fig. 16. Asymptotic behavior of the four architectures by varying p_{erratic}.

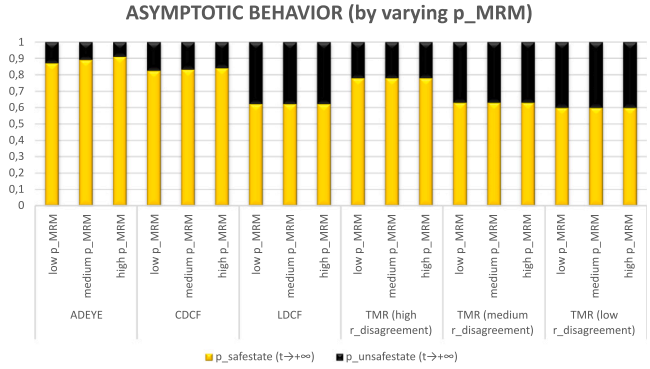

Fig. 17. Asymptotic behavior of the four architectures by varying p_{MRM}.

distinct failure manifestations. In particular, in these architectures some silent failures tend to directly activate the pre-buffered emergency maneuver, which stops the vehicle immediately and therefore causes early mission termination. For instance, in CDCF, a silent failure affecting the monitoring function can trigger the pre-buffered MRM, resulting in an immediate stop and reduced MTTF. Conversely, when failures manifest as erratic behavior, the system may execute a controlled emergency maneuver over a finite time window, slightly delaying termination and thus increasing the operational lifetime. A similar rationale applies to AD-EYE, where the increase in MTTF can be further justified by the monitoring design assumptions: the logical correctness of the fallback channel is not explicitly verified. As a consequence, an erratic failure in the second channel may remain undetected, allowing the vehicle to continue operating for longer and therefore increasing the measured MTTF. In contrast, LDCF remains nearly unchanged, suggesting that the termination time under erratic or silent failures is equivalent. Finally, TMR performance degrades with increasing p_{erratic}, with the effect becoming progressively stronger when moving from high-to medium- and especially low-disagreement configurations. This is consistent with the fact that erratic failures more easily induce channel disagreement, accelerating mission termination, whereas silent failures can be tolerated more effectively as long as at least one channel remains operational.

Regarding the asymptotic behavior (Fig. 16), for architectures with aggressive failure-handling, such as AD-EYE and CDCF, the variation is minimal: since these solutions attempt to stop at the first failure regardless of its manifestation, the long-term terminal distribution is only marginally affected. Conversely, for architectures that allow failures to accumulate over time, such as LDCF and the TMR variants, the effect of p_{erratic} becomes more pronounced. In these cases, increasing the likelihood of erratic behavior results in a higher probability of converging to an unsafe terminal state, as expected. Moreover, the sensitivity is more pronounced for LDCF than for the TMR variants: in TMR, unsafe termination is mainly associated with relatively rare voting-related events (e.g., bad majority), whose probability remains low even when p_{erratic} increases, resulting in only mild variations in the asymptotic terminal distribution.

4.6. Effect of pre-buffered MRM success probability

This subsection analyzes how varying the success probability of the pre-buffered Minimal Risk Maneuver (p_{MRM}) affects the dependability across the four architectures. Three representative values – low (0.75), medium (0.85), and high (0.95) – were considered to capture the transition from less reliable to highly effective emergency maneuvers. This parameter is only relevant when the system is unable to reach a decision, as the pre-buffered MRM provides a last-resort fallback response in such situations.

Variations in p_{MRM} have no impact on the mean time to failure for any of the considered architectures, since this parameter affects only the outcome of the pre-buffered emergency maneuver and not the time of mission termination. For this reason, the corresponding mean time to failure plot is not reported.

Conversely, the impact of p_{MRM} becomes apparent when considering asymptotic probabilistic metrics (Fig. 17). As expected, changing p_{MRM} directly affects the safe/unsafe terminal split: higher p_{MRM} increases the probability of converging to a safe terminal state, whereas lower p_{MRM} increases unsafe termination. A key observation is that the impact of this parameter is not uniform across architectures. In particular, the effect is much more visible for AD-EYE and CDCF, which rely more heavily on the pre-buffered emergency maneuver as termination mechanism. As a consequence, variations in the probability of successful MRM translate into a stronger shift in their asymptotic safe/unsafe distribution. Conversely, for architectures where termination is less directly dominated by the pre-buffered MRM, the sensitivity to p_{MRM} is noticeably weaker, resulting in only marginal changes in the terminal split.

5. Threats to validity

The validity of this study is influenced by several modeling and methodological assumptions that should be considered when interpreting the results.

First, the proposed dependability analysis relies on Stochastic Activity Networks with exponentially distributed failure times. While this assumption enables tractable analysis, it implies constant failure rates and therefore does not capture phenomena such as component aging, which may affect failure behavior over long-term operation. The adopted modeling framework further abstracts system behavior by excluding repair and maintenance actions and by not explicitly representing the complete fault–error–failure propagation chain. As a consequence, failures are modeled as direct transitions to system-level failure conditions, without capturing how faults are activated, detected,

propagate through intermediate error states, or may be mitigated before reaching a system-level outcome. As part of this abstraction, mechanisms such as recovery actions, diagnostic latency, and intermediate degraded states are not explicitly modeled. In practice, these mechanisms can influence how failures evolve and are handled over time. For example, recovery actions may allow the system to return to an operational state, diagnostic latency may delay fault detection and handling, and degraded states may introduce intermediate operating conditions (e.g., with reduced functionality or performance) before a system-level condition is reached. Overall, these modeling choices may affect both the timing and the evolution of failures, and consequently the quantitative values of the evaluated metrics. Nevertheless, the adopted framework captures the primary failure-handling mechanisms of each architecture, which are the main focus of this study. Therefore, although more detailed models could refine the quantitative results, the comparative trends observed across architectures are expected to remain representative of their underlying design characteristics.

Another aspect concerns the choice and interpretation of model parameters. Parameters were selected to represent plausible operating conditions and explored over representative ranges through a dedicated sensitivity analysis. It is worth noting that the specific parameter values adopted in this study are based on preliminary estimates and on guidelines agreed with the *Safety & Architecture* working group of *The Autonomous* (Escuela et al., 2023). While the sensitivity study confirms that the main trends remain stable across the explored ranges, some uncertainty in the exact quantitative parameter values naturally persists. The model further assumes homogeneous failure rates within each component class (complex/simple) used across all architectures. While this abstraction supports a controlled architectural comparison, it may not capture potential differences in failure behavior across architectures in real implementations. However, the conducted sensitivity analysis shows that the relative trends and ranking among architectures remain stable under uniform variations of the failure rates, supporting the robustness of the presented conclusions. In addition, simple components are assumed to experience only independent failures and to fail exclusively in a silent manner. While this is consistent with their intended design as fully verifiable and deterministic units, it does not explicitly capture additional failure behaviors that may arise in practice. Regarding generalizability, the comparison deliberately abstracts away from hardware- and software-specific implementation details, focusing on architectural principles. While this design choice enhances generality and supports architecture-level reasoning, it limits the direct applicability of the numerical results to specific platforms or products. In addition, the analysis is conducted within a specific operational design domain, namely an SAE Level 4 Highway Pilot scenario. While this enables a focused and consistent comparison, different operational contexts – such as urban driving or low-speed automated mobility applications – may involve different assumptions. As a result, the observed comparative trends should be interpreted with respect to the considered operational context, and may vary under different ODD conditions.

Finally, the conclusions drawn from this study are inherently comparative and relative in nature. The results are intended to highlight trade-offs and trends among architectural solutions rather than to provide absolute guarantees of dependability. In addition, the absence of empirical cross-validation against real systems or experimental data means that the findings should be interpreted as model-based insights rather than predictive estimates of real-world performance, and the observed quantitative differences may be influenced by the adopted modeling assumptions.

6. Conclusions

This study examined four representative fail-operational architectures using a unified SAN-based modeling framework, focusing on how their structural principles influence system behavior under failures. The analysis highlights distinct mechanisms through which redundancy, isolation, and fallback strategies shape system-level dynamics.

At the architectural level, the comparison highlights that TMR behavior is shaped by redundancy management and disagreement dynamics. Low/medium-disagreement TMR maximizes operational continuity but, by prolonging operation under degradation, increases exposure to unsafe outcomes. High-disagreement TMR is instead dominated by disagreement-driven interruptions and frequent pre-buffered MRM activations, resulting in the lowest safety, reliability, and MTTF. CDCF follows a conservative strategy driven by immediate reaction to failures, stabilizing safety at the expense of continuity. LDCF sustains operation through layered redundancy, but becomes increasingly vulnerable when failures accumulate across layers. AD-EYE shows intermediate behavior, producing a more balanced response under failure.

Sensitivity analyses refine these conclusions. Varying the fraction of independent versus common-cause failures mainly rescales the failure timescale: MTTF changes consistently across architectures, while asymptotic termination remains largely unchanged. In contrast, varying the probability of erratic behavior yields different architecture-dependent effects. In terms of MTTF, AD-EYE and CDCF increase as the probability of erratic failures grows, because silent failures more often trigger an immediate pre-buffered MRM, whereas erratic failures may lead to a controlled emergency maneuver over a finite time window. LDCF remains nearly unchanged, indicating that erratic versus silent manifestations have limited impact on its termination time. Conversely, all TMR variants degrade with increasing probability of erratic failures, as they more easily induce disagreement and accelerate mission termination. In the asymptotic regime, AD-EYE and CDCF are only marginally affected because they tend to stop at the first failure, while LDCF and TMR show a clearer shift towards unsafe termination as erratic behavior becomes more likely, since failures can accumulate over time. Finally, varying the pre-buffered MRM success probability does not affect operational longevity, but directly shifts the asymptotic safe/unsafe termination split, most visibly for architectures relying more heavily on this last-resort mechanism, such as AD-EYE and CDCF.

These results extend current understanding of fail-operational design, which has often relied on qualitative arguments or architecture-specific studies. By offering a directly comparable quantitative view, the analysis shows how design choices shape probabilistic and temporal dependability, including sensitivity to parameter uncertainty. The observed behaviors offer an evidence-based interpretation of the architectures' dependability characteristics and a coherent foundation for supporting informed reasoning on fail-operational design choices in automated driving systems. However, these results are derived from a model-based analysis and have not been empirically validated against real systems or experimental data; therefore, the observed quantitative differences should be interpreted as indicative rather than definitive, while still providing a consistent basis for comparing the relative behavior of the considered architectures.

CRediT authorship contribution statement

Manuel Drago: Writing – review & editing, Writing – original draft, Methodology, Formal analysis. **Andrea Bondavalli:** Supervision. **Paolo Lollini:** Supervision. **Moritz Antlanger:** Conceptualization. **Sascha Drenkelforth:** Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Manuel Drago reports financial support was provided by Italian Ministry of University and Research (MUR). If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

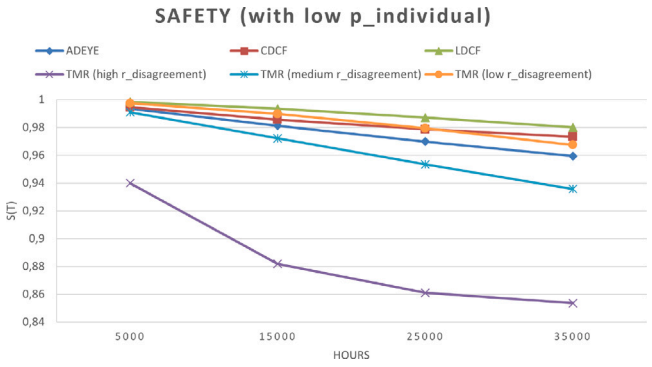

Fig. A.18. Safety of the four architectures with low $p_{\text{individual}}$ over time.

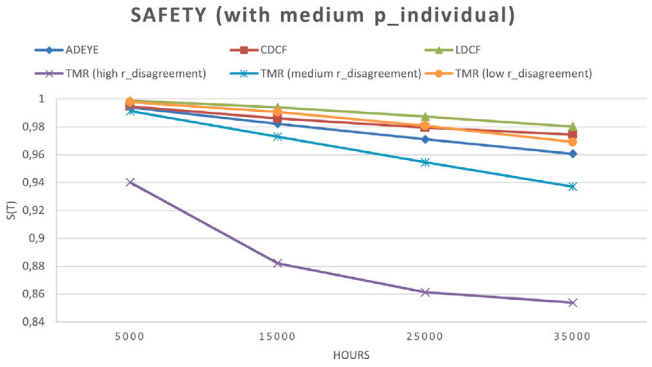

Fig. A.19. Safety of the four architectures with medium $p_{\text{individual}}$ over time.

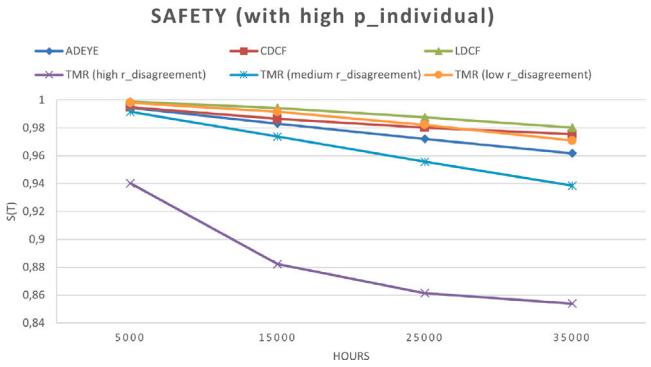

Fig. A.20. Safety of the four architectures with high $p_{\text{individual}}$ over time.

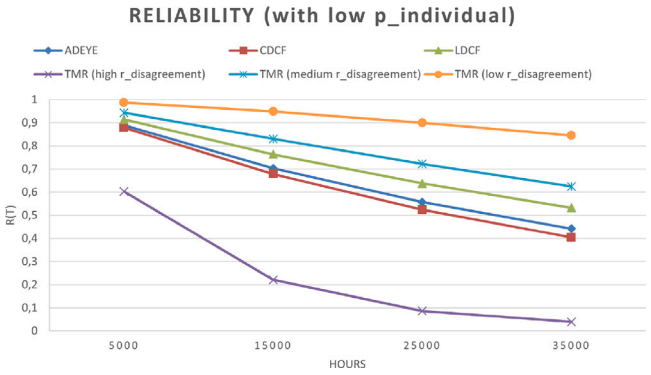

Fig. A.21. Reliability of the four architectures with low $p_{\text{individual}}$ over time.

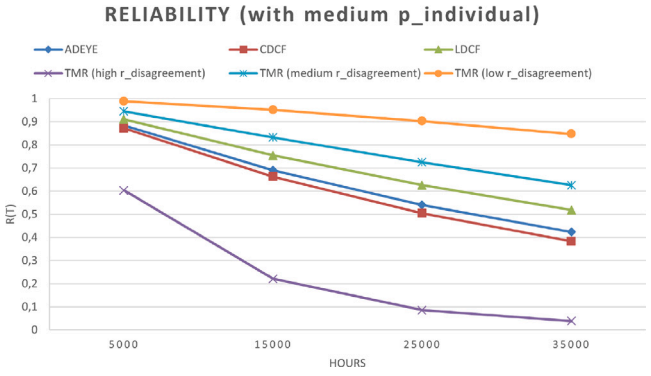

Fig. A.22. Reliability of the four architectures with medium $p_{\text{individual}}$ over time.

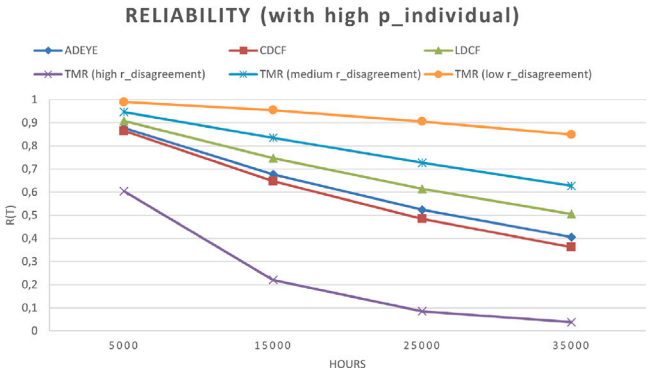

Fig. A.23. Reliability of the four architectures with high $p_{\text{individual}}$ over time.

Acknowledgment

This work was partially supported by the project S2 – PRIN 2022, Prot. no. 202297YF75, funded by the European Union under the NextGenerationEU program.

Appendix

This appendix reports additional results to complement the aggregated indicators presented in Section 4. The following figures show the evolution of safety and reliability over time under different parameter settings, providing a direct view of the transient behavior of the analyzed architectures.

For the sensitivity analysis in Section 4.4, where the independent failure proportion ($p_{\text{individual}}$) is varied, Figs. A.18–A.20 and A.21–A.23 report, respectively, safety and reliability results for low, medium, and high values of the parameter. Similarly, for Section 4.5, where the erratic behavior probability (p_{erratic}) is varied, Figs. A.24–A.26 and A.27–A.29 report the corresponding results. Finally, for Section 4.6, where the pre-buffered MRM success probability (p_{MRM}) is varied, Figs. A.30–A.32 report the corresponding safety results (no reliability plots are included, as p_{MRM} does not affect the reliability metric).

Each figure shows how the considered metric evolves over time for a given parameter setting, enabling a direct comparison across architectures under the same conditions. These results provide additional insight into how performance differences develop over the considered time horizon.

Overall, the observed trade-offs among architectures are consistent with those identified in the baseline analysis in Section 4.3, while the trends emerging as time increases are aligned with the aggregated indicators such as MTTF and asymptotic termination probabilities discussed in Sections 4.4–4.6.

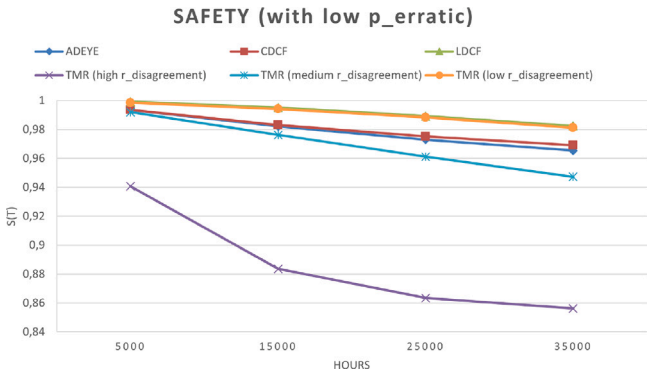

Fig. A.24. Safety of the four architectures with low p_{erratic} over time.

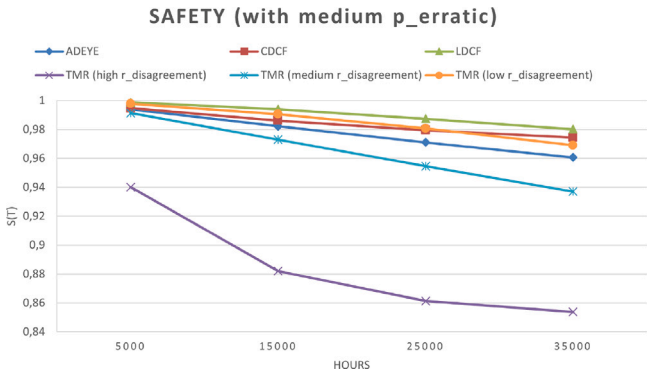

Fig. A.25. Safety of the four architectures with medium p_{erratic} over time.

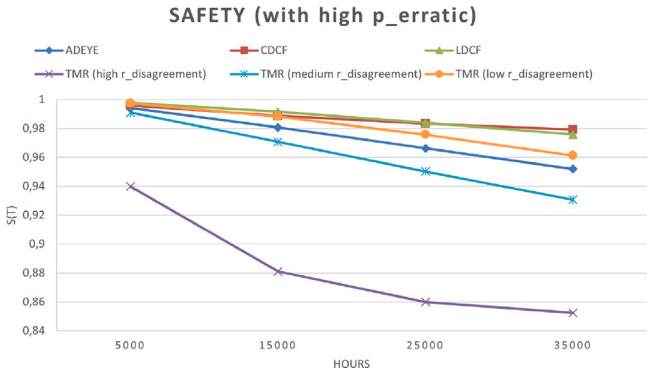

Fig. A.26. Safety of the four architectures with high p_{erratic} over time.

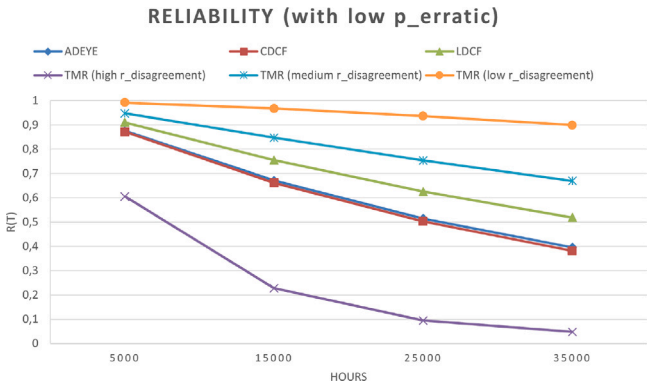

Fig. A.27. Reliability of the four architectures with low p_{erratic} over time.

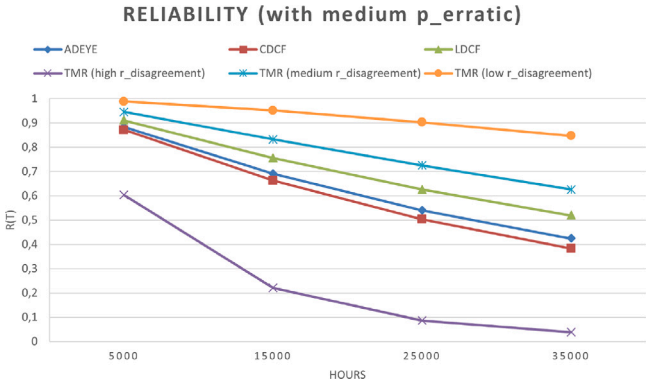

Fig. A.28. Reliability of the four architectures with medium p_{erratic} over time.

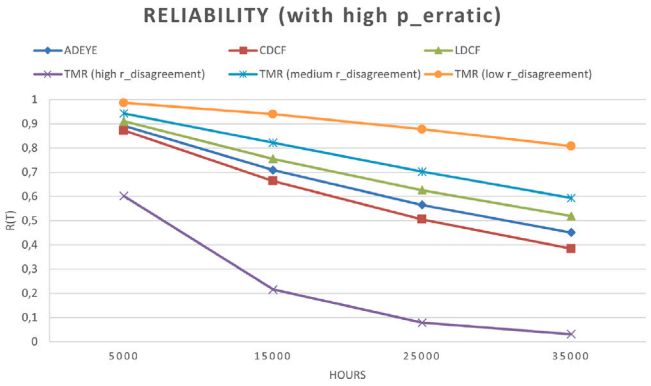

Fig. A.29. Reliability of the four architectures with high p_{erratic} over time.

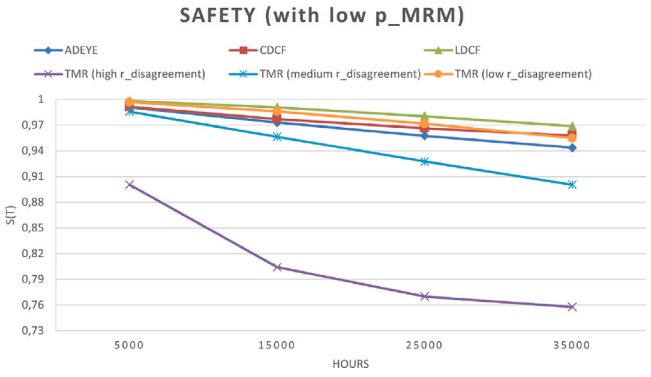

Fig. A.30. Safety of the four architectures with low p_{MRM} over time.

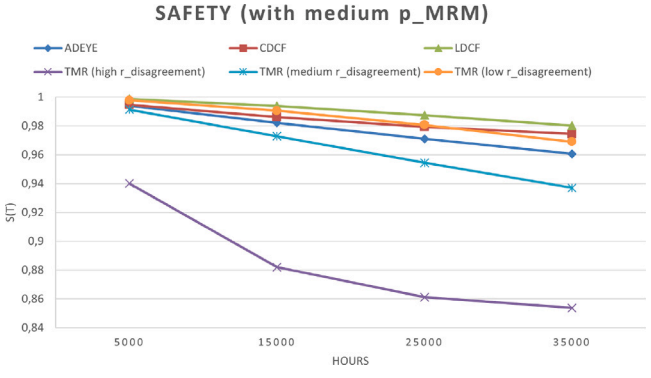

Fig. A.31. Safety of the four architectures with medium p_{MRM} over time.

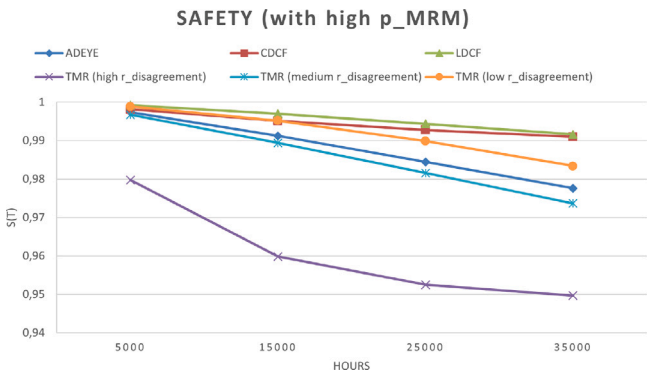

Fig. A.32. Safety of the four architectures with high p_{MRM} over time.

Data availability

The artifacts used in this study are available in the GitHub repository cited within the manuscript.

References

Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C., 2004. Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. Dependable Secur. Comput. 1 (1), 11–33. http://dx.doi.org/10.1109/TDSC.2004.2.

Boubakri, A., Gamar, S.M., 2021. A new architecture of autonomous vehicles: Redundant architecture to improve operational safety. Int. J. Robot. Control. Syst. 1 (3), 355–368. http://dx.doi.org/10.31763/ijrcs.v1i3.437, URL: https://pubs2.ascee.org/index.php/IJRCS/article/view/437.

Clark, G., Daly, D., Deavours, D.D., Derisavi, S., Doyle, J.M., Sanders, W.H., Webster, P., 2001. The Möbius modeling tool. In: Proceedings of the 9th International Workshop on Petri Nets and Performance Models. IEEE, pp. 241–250. http://dx.doi.org/10.1109/PNPM.2001.953332.

Daly, D., Deavours, D.D., Doyle, J.M., Sanders, W.H., Webster, P., 2000. Möbius: An extensible tool for performance and dependability modeling. In: International Conference on Modelling Techniques and Tools for Computer Performance Evaluation. Springer, Berlin, Heidelberg, pp. 332–336. http://dx.doi.org/10.1007/3-540-46506-5_22.

Deavours, D.D., Clark, G., Daly, D., Derisavi, S., Doyle, J.M., Sanders, W.H., Webster, P., 2002. The Möbius framework and its implementation. IEEE Trans. Softw. Eng. 28 (10), 956–969. http://dx.doi.org/10.1109/TSE.2002.1041052.

Drago, M., Bondavalli, A., Lollini, P., Niedrist, G., Antlanger, M., 2025. Quantitative comparison of system architectures for an SAE level 4 highway pilot. In: 2025 IEEE International Conference on Software Quality, Reliability and Security. QRS, pp. 612–622. http://dx.doi.org/10.1109/QRS65678.2025.00066.

Escuela, G., Stroud, N., Takenaka, K., Tiele, J.K., Dannebaum, U., Rosenbusch, J., Törngren, M., Niedrist, G., Antlanger, M., Mangold, C., Reisenberger, F., Ben Mosbeh, N., Shinde, C., Mehmed, A., Schulze, C., More, S., Fryzek, L., Storr, M., 2023. Safe automated driving: Requirements and architectures. White paper of The Autonomous Working Group Safety & Architecture, version 1.0, 01 Dec 2023. URL: https://www.the-autonomous.com/wp-content/uploads/2023/12/ta-safetyarchitecture-fullreport-a4-web-final.pdf. (Accessed 15 November 2025).

Frigerio, A., Vermeulen, B., Goossens, K.G.W., 2021. Automotive architecture topologies: Analysis for safety-critical autonomous vehicle applications. IEEE Access 9, 62837–62846. http://dx.doi.org/10.1109/ACCESS.2021.3074813.

International Organization for Standardization, 2018. ISO 26262:2018 road vehicles – functional safety.

Ishigooka, T., Honda, S., Takada, H., 2018. Cost-effective redundancy approach for fail-operational autonomous driving systems. In: 21st IEEE International Symposium on Real-Time Distributed Computing. ISORC, IEEE, Singapore, pp. 107–115. http://dx.doi.org/10.1109/ISORC.2018.00024.

Julitz, T.M., Tordeux, A., Löwer, M., 2022. Reliability of fault-tolerant system architectures for automated driving systems. In: Proceedings of the 32nd European Safety and Reliability Conference. ESREL, Dublin, Ireland, pp. 120–127.

Julitz, T.M., Tordeux, A., Löwer, M., 2023. Computer-aided design of fault-tolerant hardware architectures for autonomous driving systems. In: Proceedings of the International Conference on Engineering Design. ICED23, Bordeaux, France.

Kohn, A., 2017. Markov chain-based reliability analysis for automotive fail-operational systems. SAE Int. J. Transp. Saf. 5, 30–38. http://dx.doi.org/10.4271/2017-01-0052.

Kölbl, M., Leue, S., 2018. Automated functional safety analysis of automated driving systems. In: Howar, F., Barnat, J. (Eds.), Formal Methods for Industrial Critical Systems. Springer International Publishing, Cham, pp. 35–51.

Koopman, P., 2020. 18-642: L34 safety architectural patterns. Lecture slides, Department of Electrical & Computer Engineering, Carnegie Mellon University; https://users.ece.cmu.edu/~koopman/lectures/ece642/Xtra_SafetyArchPatterns.pdf. (Accessed 15 November 2025).

Kopetz, H., 2022. An architecture for safe driving automation. In: Principles of Systems Design – Essays Dedicated To Thomas a. Henzinger on the Occasion of His 60th Birthday. In: Lecture Notes in Computer Science, vol. 13660, Springer, pp. 61–84. http://dx.doi.org/10.1007/978-3-031-22337-2_4.

Lyons, R.E., Vanderkulk, W., 1962. The use of triple-modular redundancy to improve computer reliability. IBM J. Res. Dev. 6 (2), 200–209. http://dx.doi.org/10.1147/rd.62.0200.

Meyer, J.F., Movaghar, A., Sanders, W.H., 1985. Stochastic activity networks: Structure, behavior, and application. In: Proceedings of the IEEE International Workshop on Petri Nets and Performance Models. pp. 106–115.

Mohan, N., Törngren, M., 2019. AD-EYE: A co-simulation platform for early verification of functional safety concepts. http://dx.doi.org/10.48550/arXiv.1912.00448.

SAE International, 2021. SAE J3016: Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. SAE Standard J3016™. URL: https://www.sae.org/standards/content/j3016_202104/.

Sanders, W.H., Meyer, J.F., 2000. Stochastic activity networks: Formal definitions and concepts. In: Lectures on Formal Methods and Performance Analysis. Springer, Berlin, Heidelberg, pp. 315–343. http://dx.doi.org/10.1007/3-540-45352-0_8.

Sari, B., 2020. Fail-operational Safety Architecture for ADAS/AD Systems and a Model-driven Approach for Dependent Failure Analysis, first ed. In: Wissenschaftliche Reihe Fahrzeugtechnik Universität Stuttgart (WRFUS), Springer Vieweg, Wiesbaden, Germany, p. 147. http://dx.doi.org/10.1007/978-3-658-29422-9.

Schmid, T., Schraufstetter, S., Wagner, S., Hellhake, D., 2019. A safety argumentation for fail-operational automotive systems in compliance with ISO 26262. In: 4th International Conference on System Reliability and Safety. ICSRS, IEEE, pp. 484–493. http://dx.doi.org/10.1109/ICSRS48664.2019.8987554.

Stoffel, M., Schindewolf, M., Sax, E., 2024. On-demand triple modular redundancy for automotive applications. In: IEEE International Systems Conference (SysCon). IEEE, Montreal, Canada, pp. 1–8. http://dx.doi.org/10.1109/SYSCON60506.2024.10123456.

Trivedi, K.S., 2016. Probability and Statistics with Reliability, Queuing, and Computer Science Applications, second ed. John Wiley & Sons, Inc., Hoboken, NJ, http://dx.doi.org/10.1002/9781119285441.

Wagner, M., Ray, J., Kane, A., Koopman, P., 2017. A safety architecture for autonomous vehicles. European Patent, published 1 April 2020. URL: https://patents.google.com/patent/EP3400676B1/en.