International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(23s) 2026 pp. 1471-1489
https://doi.org/10.70917/ijcisim-2026-5706
Article

Compute Centralization as the Limiting Factor for Software-Defined Architectures in Off-Highway Equipment

Shyam Laheri Chunduri

Independent Researcher, USA

Abstract: Discussions of the software-defined vehicle (SDV) paradigm have developed almost entirely around automotive assumptions, where middleware fragmentation and connectivity availability get treated as the primary obstacles to adoption. For off-highway equipment construction, mining, and agricultural machinery, a different constraint does the actual gating: compute-architecture centralization. Whether a machine can support over-the-air (OTA) updates, on-board artificial intelligence (AI) inference, and continuous validation depends less on middleware maturity than on whether its underlying electrical architecture has crossed the threshold from domain-based electronic control units (ECUs) to a zonal or centralized compute platform. This article maps the automotive E/E architecture literature against off-highway-specific constraints, extended environmental qualification requirements, machinery functional safety standards under IEC 62061 and ISO 13849, service lives exceeding fifteen years, and remote-site connectivity limits to identify where domain architecture reaches its ceiling and centralization becomes structurally necessary rather than merely convenient. The validation-infrastructure burden of that transition, more than the compute hardware itself, is what off-highway organizations tend to underestimate. A platform-transition framework distinguishing greenfield software-defined design from legacy-fleet bridging is proposed, grounded in hardware-in-the-loop (HIL) validation-infrastructure economics rather than in architectural description alone.

Keywords: Software-defined vehicle architecture; off-highway equipment; compute centralization; zonal architecture; hardware-in-the-loop validation; over-the-air updates; functional safety; edge AI

1. Introduction

Software-defined capability has become table stakes that off-highway equipment manufacturers are now expected to deliver: over-the-air updates that reach a machine without a technician visit, on-board AI inference supporting autonomous or semi-autonomous operation, and validation pipelines built for software shipped continuously rather than at model-year intervals. Automotive engineering has spent roughly a decade working through how to build these capabilities, and its literature converges on two obstacles as the central bottlenecks: a fragmented middleware landscape limiting software portability across suppliers and connectivity infrastructure that determines how reliably updates and telemetry reach a vehicle in the field [1], [2]. Off-highway programs borrowing from this literature tend to inherit that framing without examining whether it actually fits their domain.

It does not fit cleanly. The argument developed here is that compute-architecture centralization, the shift from a collection of dedicated, domain-based ECUs to a zonal or fully centralized compute platform, is what actually gates software-defined capability in off-highway machinery, and it is a constraint off-highway programs reach later and at a higher relative cost than automotive programs do. Recent work quantifying automotive E/E architectures' centralization potential treats compute consolidation as one variable among several worth weighing [3]. In the off-highway domain, this analysis contends, it is the dominant variable because the domain's own requirements extended environmental qualification, the machinery-specific safety frameworks in IEC 62061 and ISO 13849 [4], service lives that routinely run past fifteen years, and a validation infrastructure base still built around function-isolated HIL practice

push the cost and risk of the domain-to-centralized transition well past what automotive's more mature supply chain experiences.

The stakes of this argument are practical rather than purely academic. An off-highway engineering organization deciding whether to greenlight a centralized compute platform program is, in effect, deciding how to allocate capital between two very different kinds of investment: compute hardware and middleware licensing on one side and validation infrastructure HIL benches, SIL/PIL environments, and the staffing to operate them on the other. When the architectural literature frames middleware and connectivity as the central obstacles, it implicitly steers that capital allocation decision toward hardware and software-standards spending. If validation infrastructure is instead the actual gating constraint, as this analysis argues, an organization following the automotive-derived framing risks underfunding the one capability that determines whether the resulting platform can actually be qualified for field release on schedule. This is not a hypothetical concern for organizations with decades of investment in function-isolated HIL practice, where the existing validation infrastructure was built for exactly the architecture the centralization transition is meant to replace.

The consequences of getting this sequencing wrong are not confined to a single failed program. An off-highway OEM's compute-architecture decision typically also sets the technical envelope for its Tier 1 supplier base, since suppliers building for a domain architecture invest in different tooling, different validation deliverables, and different qualification evidence than suppliers building software for a shared, centralized compute stack [5], [6]. A supplier ecosystem calibrated to one architecture does not retool overnight for the other, which means a manufacturer's centralization decision propagates outward into supplier contracts and qualification schedules, and, eventually, into how quickly a competitor's centralized-platform machine can reach the market with capabilities the manufacturer's own domain-architecture fleet cannot match. This supply-chain dimension is largely absent from the automotive-derived SDV literature, which tends to assume a supplier base already conversant with centralized, software-defined development practice.

For the purposes of this analysis, off-highway equipment covers wheeled and tracked construction machinery; mining haul trucks and loaders; and field agricultural machinery, a deliberately broad category but one that shares the constraints this article treats as decisive: extended service lives, harsh operating environments, and safety governance under the machinery-specific IEC 62061 and ISO 13849 framework rather than automotive's ISO 26262 [4] and [7]. The category excludes on-road commercial trucking, which sits closer to the automotive literature's own assumptions about supply-chain maturity and connectivity infrastructure, and it excludes marine and rail applications, whose safety and certification regimes differ enough from both automotive and mobile machinery to warrant separate treatment. Within this scope, mining and agricultural applications supply most of the published autonomy literature reviewed in Section 2.3, while construction equipment, addressed later in the limitations, remains comparatively underrepresented in that literature despite facing a similar centralization question.

Two objections deserve a direct answer before proceeding. Middleware fragmentation is genuinely present in off-highway; nothing resembling AUTOSAR Adaptive's automotive-wide adoption exists for machinery [2], and remote-site connectivity does constrain what a fielded machine can receive over cellular or satellite links. Neither is dismissed here, but neither is treated as the binding constraint either. A manufacturer can adopt middleware piecemeal, and connectivity limitations can be engineered around through staged rollout and selective data logging; neither obstacle actually blocks the start of a software-defined transition. Compute centralization does block it, because the validation infrastructure a centralized platform demands HIL rigs capable of exercising an integrated software stack rather than isolated ECU functions one at a time cannot be built incrementally alongside an existing domain architecture. It has to exist for the target architecture before that architecture can be qualified for field release.

Two contributions follow from this argument. The first identifies the specific point where domain architecture's compute and bandwidth ceiling gets reached under off-highway autonomy and edge AI workloads, treating the automotive centralization literature as a comparative baseline rather than a template to copy. The second proposes a platform-transition framework greenfield software-defined design set against legacy-fleet bridging through telematics

gateways built on validation-infrastructure economics specific to HIL, software-in-the-loop (SIL), and processor-in-the-loop (PIL) environments, rather than on the architectural description that dominates existing accounts of SDV adoption. The method is a literature-grounded comparative synthesis: the automotive E/E architecture and SDV-enabler literature are reviewed for what they establish about domain, zonal, and centralized models; that literature is then evaluated against a constraint set specific to off-highway operation; and the resulting comparison is used to identify where the two domains diverge and what that divergence implies for platform-transition strategy. Section 2 reviews the automotive E/E architecture and SDV-enabler literature and locates where it goes silent on off-highway constraints. Section 3 develops the analytical framework and constraint set applied to the architecture comparison. Section 4 presents the comparative findings on where domain architecture's ceiling sits and what the centralization threshold demands of validation infrastructure. Section 5 turns to platform-transition strategy, the middleware standardization gap, and safety and cybersecurity implications. Section 6 concludes.

2. Related Work

2.1 Automotive E/E Architecture Evolution: Domain, Zonal, and Centralized Models

Vehicle electrical and electronic systems have historically been organized around the domain architecture, in which dedicated ECUs serve each functional domain (powertrain, chassis, body, infotainment) communicating over a shared bus, typically Controller Area Network (CAN) or CAN with Flexible Data-Rate (CAN FD). Broy et al.'s foundational account of automotive software engineering traces this model back to a hardware-first design tradition, where a function and the hardware controlling it were co-specified and effectively inseparable [5]. Mature tooling and a stable supplier ecosystem followed from that tradition, but so did a structural weakness: every functional change stayed tied to a hardware qualification cycle. That coupling is exactly what strains as functional complexity grows. Bus bandwidth saturates under high-frequency sensor exchange. New features tend to bring new ECUs and bus segments with them. And functions needing to coordinate across domain boundaries pick up gateway latency that becomes structurally limiting once autonomy and machine-learning workloads require simultaneous access to sensor data spread across multiple domains.

A useful corrective comes from Wang et al.'s review of the E/E architecture literature, which pushes back on treating domain, zonal, and centralized models as a strict sequential roadmap as though every program has to pass through zonal on the way to centralized. A more accurate account evaluates domain-oriented and zone-oriented strategies in parallel against shared criteria: compute consolidation, wiring complexity, and the networking technologies each approach depends on [1]. Their survey covers automotive Ethernet as the physical-layer backbone that both zonal and centralized architectures depend on, time-sensitive networking as the scheduling discipline that gives that Ethernet backbone the deterministic latency guarantees safety-relevant functions require and service-oriented communication as the software-level pattern that lets functions discover and consume each other's data without hard-coded point-to-point wiring at the software layer, mirroring what physical Ethernet does at the hardware layer. This distinction between physical-layer and software-layer consolidation matters for the off-highway comparison developed later in this article: a machine can adopt an Ethernet backbone without also adopting service-oriented middleware, and the two investments carry different validation implications.

Mauser and Wagner extend this line of inquiry by treating centralization itself as a matter of degree rather than a binary architectural choice, identifying the system properties that determine how much centralization potential a given vehicle program actually has [3]. Their framework weighs safety-criticality separation, how cleanly a program's functions divide into safety-critical and non-safety-critical categories, against update-frequency requirements and cross-domain data access needs, on the reasoning that a program with cleanly separable criticality levels and infrequent updates gains less from centralization than one whose functions are tightly coupled and updated continuously. Industry analysis reinforces that this is a live transition rather than a settled one: recent market projections put the global share of vehicles built on genuinely zonal architecture at only around 18 percent by 2030, with domain architecture still accounting for roughly 48 percent of the market and distributed architecture for the remainder, even as the compute-unit segment serving zonal and central platforms is projected to grow at 30 to 40 percent annually against a contracting

conventional ECU market [8]. The transition, in other words, is real and directionally clear but far from complete even within automotive's more mature supply chain, a point directly relevant to any expectation that off-highway will move faster or more cheaply.

This progression is not solely an architectural story; the underlying network technology had to evolve in step with it. Controller Area Network, the bus technology that made domain architecture viable in the first place, tops out at bandwidths too low to carry raw camera or LiDAR data, which is part of why Wang et al. treat automotive Ethernet's higher bandwidth and time-sensitive networking's deterministic latency guarantees as necessary, not incidental, enablers of the zonal and centralized architectures built on top of them [1]. An off-highway machine retrofitted with a centralized compute node but still wired around a legacy CAN backbone gains little from that compute investment, because the sensor data the new compute node needs still has to reach it over a network that was never designed to carry it.

2.2 Software-Defined Vehicle Enablers in the Automotive Literature

The literature on what makes a vehicle genuinely software-defined, as opposed to merely computer-controlled, converges on a small set of enabling capabilities. Middleware standardization is treated as foundational: Kugele, Obergfell, and Sax's model-based approach to resource allocation for service-oriented automotive architectures demonstrates how platform-independent service models can be synthesized against hardware constraints and then transitioned into AUTOSAR Adaptive implementations, bridging early architectural design with production deployment [6]. Their method uses design-space exploration to map candidate service deployments onto available compute resources while respecting processor utilization, memory, and safety constraints, which gives program teams a way to evaluate deployment options analytically before committing to a specific hardware target. This kind of tooling presumes a middleware target that is stable enough to design against, which is what the AUTOSAR Adaptive Platform provides for automotive [2] and what off-highway currently lacks.

Over-the-air update capability is treated as the operational expression of the SDV architecture rather than an add-on feature, and its correctness requirements draw close scrutiny in the literature specifically because of what happens when they fail. Mahmood's, Nguyen's, and Shaikh's systematic threat assessment of automotive OTA update mechanisms documents the breadth of the attack surface an OTA-capable vehicle exposes, spanning update-package integrity verification, rollback-mechanism exploitation, and man-in-the-middle interception of the update channel itself, and argues that security testing has to be built into the update pipeline design rather than layered on afterward [9]. Pascale et al.'s work on intrusion detection for connected vehicles addresses the adjacent concern of monitoring the in-vehicle network once an OTA channel exists, using network-traffic-pattern analysis to flag anomalous behavior on the in-vehicle bus and treating detection as a necessary complement to update-channel hardening rather than a substitute for it [10]. Taken together, this literature establishes that OTA capability is inseparable from the cybersecurity architecture built around it, a point that carries directly into the off-highway discussion in Section 5.3.

Validation practice is the third enabler, and it is where the automotive literature has moved furthest from the milestone-gated model that off-highway validation teams, including HIL organizations built around dedicated ECU qualification, still largely operate under. Abboush, Knieke, and Rausch's continuous integration framework for automotive software validation pairs hardware-in-the-loop testing with automated failure classification, combining a long short-term memory model trained to recognize known fault signatures with a clustering approach for previously unseen faults and reporting a fault-classification F1 score of 91.85 percent for the combined approach [11]. Their broader point is that continuous validation pipelines can absorb the testing burden that milestone-gated HIL campaigns push to the end of a release cycle, since automated triage lets a small human review team keep pace with a much larger volume of automatically generated test results. Mauser et al.'s systematic literature review of mixed-criticality software architectures for centralized high-performance-compute platforms in software-defined vehicles surveys how safety-critical and non-safety functions can be integrated on shared silicon while preserving the isolation guarantees a domain architecture achieves for free through physical ECU separation, extracting from the literature a set of functional domains, constraints, and enabling technologies relevant to mixed-criticality integration and proposing an exemplary

architecture pattern for microprocessor-based system-on-chip implementations that gives practitioners a starting point rather than a green-field design exercise [12].

OTA capability changes more than the software delivery mechanism; it changes the field service relationship. Mahmood, Nguyen, and Shaikh's threat assessment of automotive OTA mechanisms treats the update channel as an attack surface precisely because it replaces what used to require a technician's physical presence and a diagnostic tool plugged into the vehicle with a remote software transaction that must be trusted without that physical verification step [9]. For off-highway equipment, this substitution is more consequential than in automotive, because remote job sites, the underground mine, the row-crop field, and the highway construction corridor are exactly the settings where sending a technician is most expensive and least timely, which is also why an off-highway OTA failure carries a different cost profile than an automotive one: a stalled passenger vehicle can usually reach a dealership under its own power, while a stalled haul truck at a remote pit face cannot.

2.3 The Off-Highway Domain Literature Gap

What this automotive-centric literature does not address is the off-highway operating envelope. Machinery functional safety is governed by IEC 62061 and ISO 13849 rather than the automotive-specific ISO 26262 [4], and the translation between the two frameworks is not mechanical: a hazard analysis conducted for an automotive perception function does not transfer to an autonomous loader or mining truck operating near personnel without being redone against the machinery-specific standard. Emerging frameworks for AI-related safety properties in vehicles, such as ISO/PAS 8800, begin to address perception-system safety questions, including distribution shift between training and deployment conditions and operational-design-domain specification that matters equally for off-highway autonomy but was written with the automotive operational context in mind and requires deliberate adaptation before it can be applied to a machinery hazard analysis [13].

The off-highway autonomy literature that does exist tends to focus on the autonomy function itself rather than the underlying compute architecture that makes it possible. Chen et al.'s account of cooperative autonomous mining operations describes a parallel-intelligence framework that coordinates fleets of autonomous haul trucks across a mine site, allocating routes and sequencing loading operations across the fleet in a way that treats each truck as a node in a larger coordination problem rather than an isolated autonomous agent, but the on-board compute platform enabling each truck's local perception and control enters their account as a given rather than as a design variable with its own centralization threshold [14]. Aydemir and Unel's work on motion planning and path following for a full-scale mining truck and trailer system addresses the control problem in comparable depth, formulating the trailer's kinematic constraints explicitly and deriving a path-following controller that accounts for the articulation between truck and trailer during reversing maneuvers but does not engage with how the underlying E/E architecture constrains what real-time perception and control workloads the vehicle can actually run [15].

Essien and Frimpong's review of three-dimensional object detection systems for autonomous truck navigation in underground mines surveys the perception algorithms in detail, covering LiDAR-based, camera-based, and sensor-fusion approaches to detecting obstacles and other vehicles in the confined, poorly lit conditions of an underground mine while treating the compute platform they run on as largely interchangeable across implementations [16]. Wu et al.'s review of navigation technologies for autonomous field agricultural machinery follows a similar pattern, cataloguing GNSS- and RTK-based positioning approaches, machine-vision-based row and headland detection, and path-planning algorithms for field coverage, without addressing whether the machine's electrical architecture can support running several of these functions simultaneously [17]. This is the gap this article addresses: none of the off-highway autonomy literature engages directly with the compute-centralization question that the automotive SDV literature has begun to formalize, even though the autonomy functions these papers describe are exactly the workloads that make centralization necessary in the first place.

A pattern is visible across all four of these off-highway autonomy papers: each treats its specific perception, planning, or coordination problem as the unit of analysis, and each implicitly assumes that whatever compute platform the vehicle carries can support the algorithm being proposed. None asks what happens when a mine site or a field

operation wants to run more than one of these capabilities on the same machine simultaneously, which is precisely the question Section 4.1 takes up. This is not a criticism of the individual papers, whose scope is appropriately narrow for their own research questions; it is evidence that the compute-architecture question sits in a genuine blind spot between the automotive SDV literature, which addresses architecture but not off-highway constraints, and the off-highway autonomy literature, which addresses off-highway constraints but not architecture.

3. Analytical Framework

3.1 Comparative Evaluation Criteria

To evaluate where domain, zonal, and centralized architectures diverge in their suitability for off-highway software-defined capability, this analysis uses five criteria drawn from the automotive E/E architecture literature and adapted to the off-highway context: compute consolidation capacity, harness and wiring complexity, fault isolation characteristics, environmental qualification burden, and validation infrastructure scaling requirement. Each is treated in turn below, before the constraint set specific to off-highway operation is developed in Section 3.2 and the two are brought together in the comparison presented in Table 1.

Compute consolidation capacity refers to how much processing power is available to a single function or fused sensor stream. In a domain architecture, this is bounded by the microcontroller sized to that function's specific requirements at design time, which is efficient for a fixed, well-understood workload but leaves no headroom for a perception or autonomy function whose computational demands were not anticipated when the ECU was specified. Harness and wiring complexity is a direct cost and reliability driver, and one where off-highway machines are structurally worse positioned than passenger vehicles: a large wheel loader's or mining truck's harness runs considerably longer distances than an automotive harness, through a more mechanically hostile environment, which makes every point-to-point connection a larger absolute liability. Fault isolation characteristics describe how a software or hardware fault in one function propagates, or fails to propagate, to others sharing the same compute resource; a domain architecture gets strong fault isolation essentially for free, as a byproduct of physical ECU separation, whereas a centralized platform has to engineer that isolation deliberately.

Environmental qualification burden refers to what re-engineering a given compute platform requires to meet off-highway durability requirements, and it interacts directly with compute consolidation: concentrating more function value onto fewer physical nodes means each node's environmental qualification failure has proportionally larger consequences. Validation infrastructure scaling requirement, the fifth criterion, refers to what a HIL, SIL, or PIL environment must be able to exercise to qualify software running on a given architecture for field release, and it is treated in this analysis as the criterion that matters most for the off-highway centralization decision, a claim developed at length below.

These five criteria do not operate independently of one another; an architecture that scores well on compute consolidation typically scores worse on fault isolation, and an architecture that scores well on harness simplicity typically concentrates more qualification risk onto fewer nodes. Table 1 presents the resulting comparison across the three architecture types, but the table's flat structure understates how strongly the criteria interact for any single program: a manufacturer weighing a centralized platform is not choosing a favorable score on one criterion in isolation but accepting a specific bundle of trade-offs across all five simultaneously, several of which (fault isolation and validation-infrastructure scaling in particular) compound rather than offset each other.

Table 1 is intended as a comparative reference rather than a scoring rubric that mechanically outputs an architecture recommendation; the relative weight an organization places on each criterion depends on its own product mix, its existing validation infrastructure base, and its service life and regulatory obligations, all of which vary enough across off-highway manufacturers that no single weighting scheme generalizes across the industry. What the table does establish reliably is the direction of each trade-off, which is sufficient for the argument this article develops even without a universal weighting scheme.

Criterion	Domain Architecture	Zonal Architecture	Centralized Architecture
Compute consolidation capacity	Low one function per ECU	Moderate regional consolidation at zone controllers	High single or few high-performance compute nodes
Harness / wiring complexity	High-extensive point-to-point wiring	Reduced zone-local wiring plus Ethernet backbone	Lowest minimal long-haul wiring
Fault isolation	Strong physical ECU separation	Moderate zone-level isolation	Requires explicit hypervisor/OS-level isolation
Environmental qualification burden	Distributed each ECU independently qualified	Concentrated at zone controllers	Concentrated on one or few high-value nodes
Validation infrastructure requirement	Function-specific HIL benches	Zone-integration HIL capability	Full-stack integration HIL capability (highest burden)

Table 1. Comparative evaluation of domain, zonal, and centralized E/E architectures against off-highway-relevant criteria [1], [3], [5].

The last of these criteria is where this framework departs most from the automotive-centric literature it draws on. Existing accounts of centralization potential treat validation as one consideration among the system properties that determine whether a given architecture is worth adopting [3] but do not treat validation infrastructure cost as a primary driver of the adoption decision itself. This analysis treats it as exactly that, for a reason grounded in off-highway HIL practice specifically: a domain architecture can be validated function by function on relatively modest, single-purpose HIL benches because each ECU's software can be qualified largely in isolation from the others. A centralized or zonal platform cannot be validated this way, because scheduling contention, shared-resource interference, and inter-process communication behavior under load are integration-level properties that only manifest when the full software stack runs together on representative hardware.

The economics of this difference are worth making explicit. An off-highway organization that has invested for decades in a fleet of function-specific HIL benches, as is typical of established heavy-equipment engineering organizations supporting multiple product lines across several countries, faces a validation infrastructure replacement, not merely an expansion, when it adopts a centralized architecture, and that replacement cost belongs in the centralization decision from the outset rather than being discovered during the first integration test campaign. The staffing implications compound the capital cost: engineers experienced in qualifying an individual ECU's software against a fixed hardware target are not automatically equipped to diagnose scheduling contention or shared-resource interference in an integrated multi-function platform, which is a different diagnostic skill set built around observing emergent, timing-dependent behavior rather than verifying a single function's correctness in isolation. Organizations that treat the centralization decision as primarily a hardware-selection exercise risk underestimating both the capital and the staffing-retraining cost of the validation-infrastructure side of the transition.

A concrete illustration clarifies the mechanism. Consider an off-highway manufacturer maintaining, as many established heavy-equipment engineering organizations do, a fleet of ECU-specific HIL benches built up over multiple product generations. Adopting a centralized platform does not let that manufacturer repurpose those benches for the new architecture's validation needs, because the benches were built to exercise one function's software against one ECU's I/O profile, not to reproduce the scheduling and resource-contention behavior of an integrated multi-function stack. The market data on compute-unit growth cited above a projected 30 to 40 percent annual growth in the zonal- and central-compute-unit segment against a contracting conventional ECU market [8] describes hardware demand but

says nothing about the validation-infrastructure investment that has to accompany it, which is exactly the gap this analysis argues the existing literature leaves unaddressed.

3.2 Off-Highway Constraint Set

Four constraints specific to the off-highway domain modify how the criteria above apply relative to automotive. First, environmental qualification: construction, mining, and agricultural machines operate under sustained mechanical shock and vibration, wider thermal ranges, and higher ingress of dust and moisture than automotive test standards anticipate, and electromagnetic interference from high-current hydraulic and electric-drive systems adds a further qualification burden that automotive-qualified compute platforms do not automatically satisfy. A centralized compute platform concentrates this qualification risk onto a single node rather than distributing it across many smaller, individually simpler ECUs, which raises the stakes of getting environmental qualification right the first time a single-node qualification failure on a centralized platform is a program-level risk in a way that a single ECU's qualification failure in a domain architecture, however costly to remediate, generally is not.

Second, the machinery functional safety framework itself. IEC 62061 and ISO 13849 [4] structure hazard analysis and safety-integrity-level assignment differently from ISO 26262 [7], and a centralized architecture that must host functions of different safety-integrity levels on shared compute resources needs isolation mechanisms at the hypervisor or operating-system level that are explicitly evaluated against the machinery standard's own architecture requirements, not carried over from an automotive ASIL decomposition exercise. This distinction matters concretely for engineering teams with automotive-adjacent experience: a hypervisor partitioning scheme validated against ISO 26262's ASIL decomposition rules does not automatically satisfy the performance-level architecture categories that ISO 13849 defines, and treating the two as interchangeable is itself a source of certification risk on a centralized off-highway platform.

Third, service life: off-highway machines routinely remain in field service for ten to fifteen years or longer, meaning a compute platform selected today must remain serviceable, software-updatable, and cyber-secure well beyond the timeframe automotive platform-refresh cycles anticipate, which raises the long-term cost of choosing a compute architecture that becomes difficult to source or maintain. This constraint interacts with compute consolidation in a way the automotive literature does not have to consider at the same scale: a centralized platform chosen today, with components sourced from suppliers whose own product roadmaps may not extend fifteen years, carries an obsolescence-management burden that a domain architecture's more modest, more readily second-sourced ECUs do not carry to the same degree.

This obsolescence exposure is not merely a component-sourcing inconvenience. A compute platform's software toolchain, compiler support, and security-patch lifecycle are all tied to the same supplier roadmap as the hardware itself, and a supplier's decision to end-of-life a system-on-chip family after eight or ten years is a routine decision in consumer and even automotive electronics that leaves an off-highway program needing security patches and feature updates for a platform its own supplier no longer actively maintains. A domain architecture's simpler, more standardized ECUs are more readily replaced with a functionally equivalent part from a different supplier when this happens; a centralized platform's software stack is typically far more tightly coupled to its specific compute hardware, which makes an equivalent substitution correspondingly harder.

Fourth, connectivity: off-highway machines frequently operate at remote job sites with intermittent cellular coverage or satellite-only connectivity, which constrains how OTA delivery and field-data collection can be designed regardless of the compute architecture chosen and is treated in this analysis as a real but independently addressable constraint through staged rollout design and selective, uncertainty-triggered data logging rather than as evidence against centralization itself. A centralized platform does not make this constraint worse; if anything, consolidating the update-management and data-logging logic onto fewer compute nodes makes it easier to implement the kind of selective, bandwidth-aware logging that intermittent connectivity requires, since that logic does not have to be replicated and coordinated across many separate domain ECUs.

These four constraints do not act in isolation on the centralization decision; they compound specifically through the validation-infrastructure criterion identified in Section 3.1. Extended environmental qualification and long service life both raise the cost of getting a centralized platform's initial qualification wrong, since a fielded correction is harder to execute at scale and over a longer horizon than in automotive's shorter refresh cycles. The machinery safety framework raises the technical bar for the isolation mechanisms a centralized platform must implement before it can be qualified at all. And the connectivity constraint, while independently addressable, still determines how much of the validation and monitoring burden has to be handled on-machine rather than offloaded to continuous cloud-based analysis, which is a validation-infrastructure question in its own right. None of these four constraints, taken alone, would be sufficient to make compute centralization the dominant consideration this article argues it is; taken together, and mapped against the validation-infrastructure criterion specifically, they are.

Taken as a set, these four constraints reinforce a point implicit throughout this framework: off-highway equipment is not simply automotive equipment operating in a harsher environment but a domain whose safety governance, ownership economics, and duty cycles differ enough from automotive's own assumptions that a compute-architecture strategy imported wholesale from automotive practice will systematically underweight exactly the risks of qualification, durability, multi-decade support, and validation-infrastructure scaling that matter most for a successful transition.

4. Comparative Findings

4.1 Where Domain Architecture Reaches Its Ceiling Under Off-Highway Autonomy and Edge AI Workloads

The off-highway autonomy literature reviewed in Section 2.3 converges, without stating it explicitly, on workloads that a domain architecture cannot support past a certain complexity threshold. Consider what an autonomous hauling operation of the kind Chen et al. describe actually requires of the on-board platform: fleet-coordination logic consuming position and status data from the truck's own systems, a perception stack processing camera and LiDAR streams to detect obstacles and other vehicles, a path-planning function consuming both the perception output and the fleet-coordination instructions, and a telemetry-logging function capturing operational data for both safety review and model improvement [14]. In a domain architecture, these would be split across perception, powertrain, and communication domains, each behind its own gateway, and the fleet-coordination and path-planning functions specifically need low-latency, high-bandwidth access to the perception output that gateway-mediated cross-domain communication was not designed to provide at the required update rate.

Real-time path following and reversing for a mining truck and trailer combination, as Aydemir and Unel's control formulation requires, depends on tight-loop access to positioning and steering-actuation data that a domain gateway's added latency directly works against, since the controller has to react to the trailer's articulation angle within a control loop whose timing budget leaves little room for gateway-induced delay [15]. The three-dimensional object detection systems Essien and Frimpong survey for underground mine navigation and the positioning and obstacle-avoidance approaches Wu et al. catalogue for field agricultural machinery both assume a compute platform capable of running perception inference at update rates the function requires without competing for bandwidth against unrelated domain traffic [16], [17]. Edge-based object detection architectures developed for on-road autonomous vehicles, which offload part of the inference pipeline to edge compute nodes to meet real-time latency budgets while retaining detection accuracy comparable to cloud-based processing, illustrate the same underlying requirement in a different domain: perception inference of this kind needs dedicated, low-latency compute access that a gateway-mediated domain architecture is not built to provide [18].

None of these workloads is intrinsically incompatible with a domain architecture in isolation; a single perception function can be given its own dedicated, sufficiently powerful ECU, and off-highway machines have shipped exactly this kind of single-function automated assist for years. What breaks down is running several such functions together, because the domain architecture's gateway-mediated cross-domain communication was never

designed for the sustained, high-bandwidth, low-latency sensor fusion that combined perception, autonomy, and continuous validation logging require simultaneously. This is the ceiling: not any single autonomy function, but the combination of functions an off-highway machine needs to run concurrently once it moves beyond a single automated assist feature toward genuine autonomous or semi-autonomous operation.

A parallel example from field agriculture illustrates that this ceiling is not specific to mining. A machine performing GNSS- and RTK-based autonomous guidance while also running machine-vision-based row detection, as the systems Wu et al. survey do individually, could in principle host each function on its own dedicated ECU in a domain architecture [17]. Layering a third capability on top continuous validation logging that captures guidance and detection performance across a full growing season for later model refinement reintroduces the same cross-domain bandwidth and gateway-latency problem described above for mining, because the logging function needs simultaneous access to both the guidance system's positioning stream and the vision system's detection output at a resolution and rate a domain gateway was not built to sustain. The specific functions differ between mining and agriculture; the architectural ceiling they run into does not.

Construction equipment, treated only briefly in the off-highway autonomy literature reviewed in Section 2.3, faces a structurally similar ceiling for a different reason: a construction site's semi-autonomous grading or compaction equipment must coordinate with other machines and personnel operating in close proximity, which raises the same combined-workload demand perception, coordination, and continuous safety-relevant logging running simultaneously even though the published literature quantifying this specific workload combination for construction machinery remains sparse relative to mining and agriculture.

The economic stakes of this ceiling are easiest to see in reverse: an off-highway manufacturer that ships a domain-architecture machine advertised as autonomy-ready, only to discover during field trials that its gateway architecture cannot sustain simultaneous perception, coordination, and logging traffic at the required rate, faces a costly late-stage redesign rather than a planned architecture decision. This is not a hypothetical risk; it is the direct consequence of treating each autonomy capability in Section 2.3's literature as independently addable to an existing platform rather than as a bundle whose combined bandwidth and latency demands should be assessed against the target architecture before the program commits to it.

4.2 The Centralization Threshold

The threshold at which zonal or centralized compute becomes necessary, rather than merely preferable, is reached when an off-highway machine requires two or more of the following simultaneously: on-board AI inference across fused multi-sensor input, OTA-deliverable software updates to the functions consuming that inference, and continuous (rather than milestone-gated) validation of the combined software stack. Figure 1 sets out the decision logic explicitly. Any one of these three capabilities can, with sufficient engineering effort, be layered onto a domain architecture. Automated assist features have shipped on domain-architecture machines for years, and telematics gateways already deliver limited configuration updates to legacy fleets without requiring a compute architecture change, a point developed further in Section 5.1. It is the combination that a domain architecture cannot sustain, because each additional capability compounds the cross-domain data access and validation infrastructure demands the others already strain: adding OTA delivery to a machine already running fused-sensor AI inference means the update pipeline now has to account for every possible combination of software version and inference-model version in the field, and adding continuous validation on top of that means the HIL infrastructure has to exercise that combinatorial space on an ongoing basis rather than at discrete release milestones.

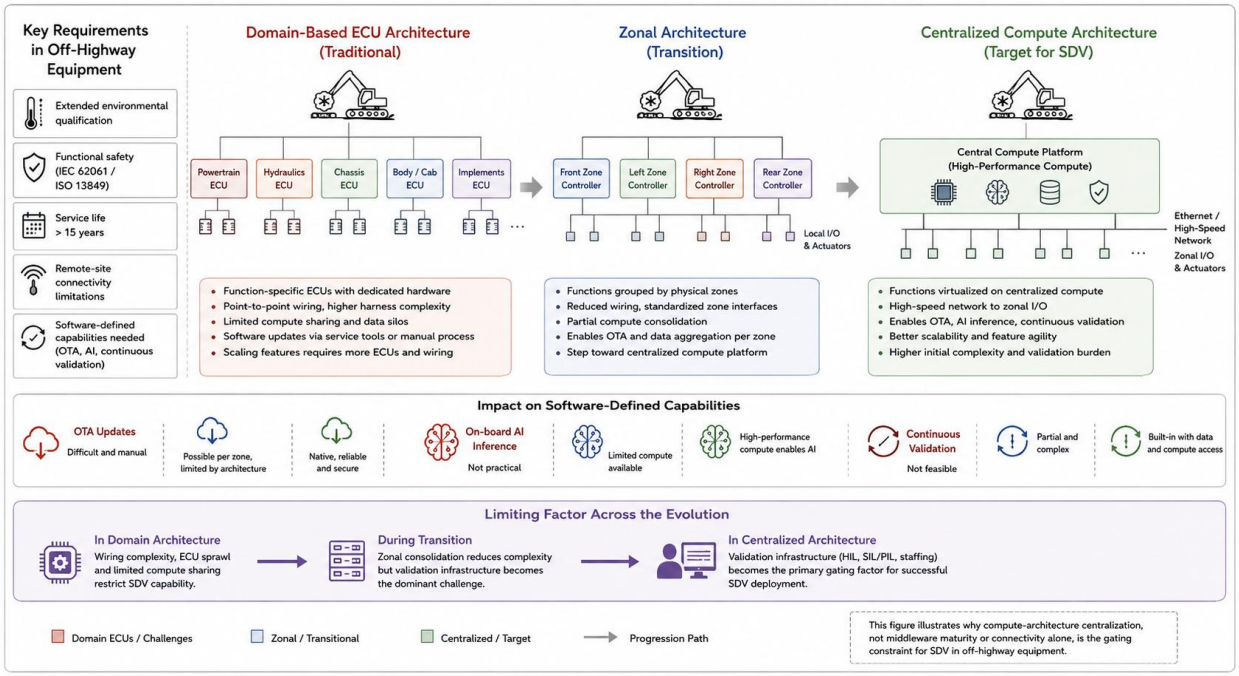

Figure 1. Decision logic for the compute-centralization threshold in off-highway equipment.

Read left to right, Figure 1 starts from a machine's stated functional requirements and asks a single branching question: Does meeting those requirements demand two or more of onboard AI inference, OTA-deliverable updates, and continuous validation running together? A negative answer keeps the program on a domain architecture where the existing validation and supply-chain practice already fits. A positive answer routes the program through the centralization threshold and directly into the validation-infrastructure scaling requirement that Section 4.3 details, which is the point in the diagram this article argues off-highway programs most often underestimate, because the diagram's earlier boxes look like an architecture decision while its later boxes are really a validation-infrastructure budgeting decision in disguise.

Mauser and Wagner's framing of centralization as a matter of degree, driven by specific system properties rather than a single threshold, is useful here but requires adaptation for the off-highway case [3]: where their analysis treats safety-criticality separation and update-frequency requirements as properties to be weighed, this analysis argues that for off-highway equipment specifically, the validation-infrastructure property dominates the others because it is the one an established off-highway engineering organization is least prepared to absorb incrementally. An automotive OEM building a new compute platform can often build new HIL infrastructure alongside the platform program from the outset, particularly at OEMs with large, greenfield-scale integration budgets and market growth data suggesting central compute unit adoption is accelerating [8]. An off-highway organization with decades of investment in domain-architecture-specific, function-isolated HIL benches faces a harder choice: it must either sustain two parallel validation infrastructures during a multi-year transition or accept a validation-coverage gap on the legacy domain fleet while capital is redirected toward centralized-platform HIL capability.

This choice plays out differently depending on organizational scale. A large, multi-product-line off-highway manufacturer has more capacity to sustain two parallel validation infrastructures during a transition, spreading the fixed cost of new integration-capable HIL benches across several concurrent platform programs, but also has a much larger existing fleet of function-isolated benches to eventually retire or repurpose. A smaller manufacturer or a single-product-line division faces a starker version of the same choice: the capital cost of new HIL infrastructure is harder to amortize across fewer concurrent programs, which can make the bridging strategy discussed in Section 5.1 more attractive as an interim step even when a greenfield centralized platform is the eventual goal. This is the centralization

threshold restated in economic rather than purely architectural terms, and it is the reason this article treats validation infrastructure, not compute hardware selection, as the actual limiting factor.

An organization that treats the three capabilities in Figure 1's decision logic as independently adoptable, rather than jointly gating the centralization threshold, risks a specific and avoidable failure mode: committing to OTA-deliverable updates and on-board AI inference on a domain-architecture machine, discovering during integration testing that the resulting cross-domain data-access and update-combinatorics burden exceeds what the existing gateway architecture and HIL infrastructure can support, and only then facing the centralization decision under schedule pressure rather than as a deliberate, budgeted program choice. This sequencing failure is precisely what the decision logic in Figure 1 is meant to make explicit and preemptable.

4.3 Validation Infrastructure Implications

The model-in-the-loop, software-in-the-loop, processor-in-the-loop, and hardware-in-the-loop progression that has structured embedded controls validation since well before the SDV era [19] does not change in name once a centralized platform is introduced, but it changes materially in what each stage must exercise, as summarized in Figure 2. At the model-in-the-loop stage, plant models of the machine's powertrain, hydraulics, and kinematics remain the lowest-cost venue for establishing functional logic, and this does not change under centralization: the physics being modeled is the same regardless of what compute platform will eventually run the control algorithm. Software-in-the-loop, which moves compiled production code into the simulation loop, begins to diverge once the production code in question is not a single function's binary but a shared software stack running under an operating system and middleware layer whose scheduling behavior the simulation must also represent a SIL environment built for a domain architecture's single-function binaries has no equivalent need to model an operating system scheduler at all, since there is no scheduling contention to represent.

The practical consequence for an off-highway validation organization is that SIL testing under centralization requires modeling operating-system scheduling behavior that a domain-architecture SIL environment could safely ignore, since a domain ECU's single-function binary has no scheduler to contend with in the first place. Building this modeling capability is not simply a matter of more powerful simulation hardware; it requires validation engineers to understand and represent middleware and operating-system behavior that previously sat entirely outside the scope of function-level software validation.

Processor-in-the-loop, which places that code on representative target hardware while keeping the plant in simulation, is where interrupt latency and scheduling-contention effects specific to a centralized, multi-function compute node first become observable, effects that a domain architecture's single-function ECUs do not exhibit because each ECU has no competing workload to contend with. This is a genuinely new validation activity for an organization moving from domain to centralized architecture, not merely a scaled-up version of an activity it already performs, because PIL testing against a domain architecture's ECUs has traditionally had little reason to focus on interrupt-latency variability under contention.

Building the capability to observe these effects typically means the validation organization must invest in target hardware that matches the production centralized platform closely enough to reproduce its real interrupt and scheduling behavior, rather than a generic development board a requirement that a domain architecture's simpler, single-function ECUs rarely impose to the same degree, since a domain ECU's interrupt behavior is far less sensitive to what else happens to be running on the same silicon at the same time.

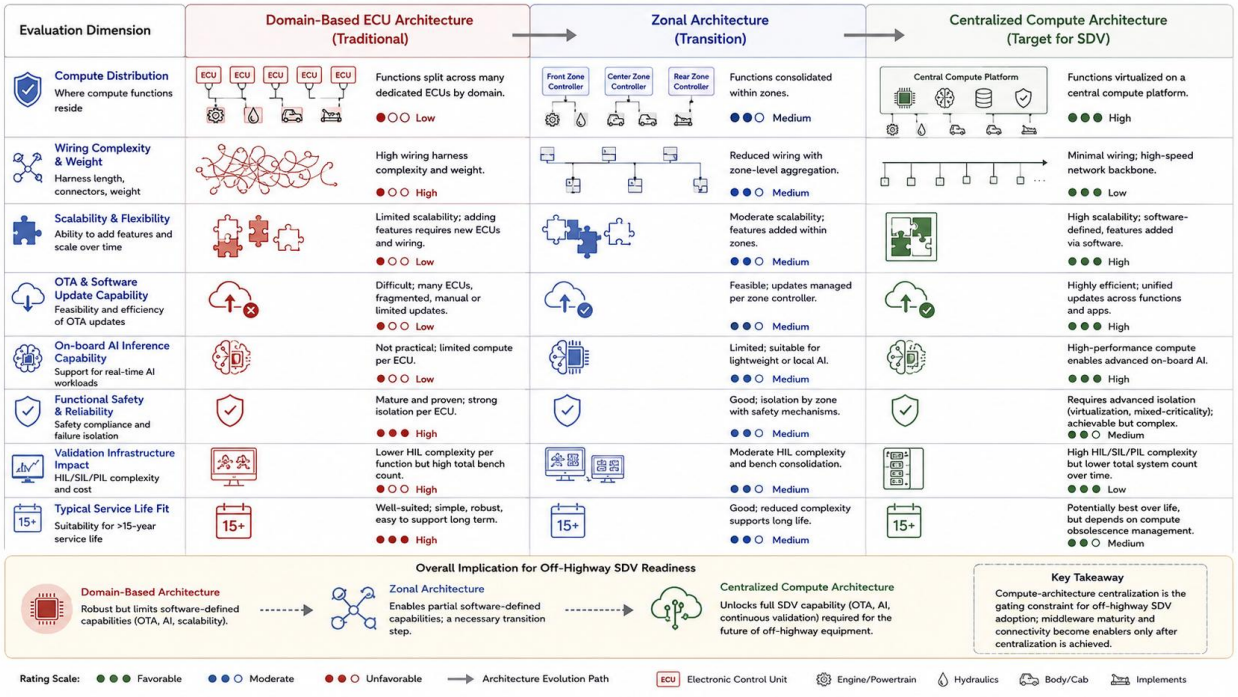

Figure 2. MIL/SIL/PIL/HIL validation hierarchy adapted to the centralized SDV context, showing what changes at each stage relative to domain-architecture practice.

Hardware-in-the-loop is where the difference is most consequential for an off-highway validation organization built around domain-architecture practice. A HIL environment validating a centralized platform must exercise the integrated system's multiple functions, the operating system, the middleware, and their combined scheduling and inter-process communication behavior because scheduling contention under load is an integration-level property that cannot be observed by testing functions in isolation, however thoroughly each is validated individually. Abboush, Knieke, and Rausch's continuous integration approach, which pairs HIL testing with automated fault classification, achieving an F1-score of 91.85 percent [11], illustrates one route to managing the resulting testing burden: their finding is not that HIL testing becomes unnecessary under continuous delivery but that automated failure triage becomes necessary to keep pace with a software stack that is changing continuously rather than at release-gated intervals. Khan and Vijay's scenario-based validation methodology for safety of the intended functionality, developed for an automotive intelligent speed assistance case study, offers a template for structuring this kind of validation systematically, generating synthetic test scenarios from hazard and risk analysis using a tree diagram decomposition of triggering conditions, rather than relying on exhaustive field-condition enumeration that transfers directly to an off-highway centralized-platform HIL program facing the same combinatorial testing problem [20]. What none of this validation literature resolves, and what an off-highway organization must resolve for itself, is the capital and staffing cost of building integration-capable HIL infrastructure in the first place, on a timeline that keeps pace with the compute-platform program rather than trailing it.

Khan and Vijay's tree-diagram decomposition of triggering conditions offers a systematic way to generate the combinatorial test scenarios a centralized platform's HIL program needs, but generating the scenarios is only part of the cost; executing them requires HIL rigs capable of running the integrated software stack under realistic timing conditions, rig time that has to be scheduled and budgeted the same way compute hardware procurement is, and validation engineers retrained in the diagnostic skills described in Section 3.1 [20]. An off-highway organization that budgets the compute-platform program's hardware and software costs without budgeting this HIL capital and staffing timeline on an equally explicit basis is, in effect, deferring a cost it will still have to pay, just later and under more schedule pressure.

Taken together, the four validation stages surveyed above describe a consistent pattern: the closer a validation activity sits to representative production hardware and an integrated software stack, the more it diverges from what a domain-architecture validation organization already knows how to do, and the more capital and retraining it demands. An off-highway organization sequencing its own transition would do well to treat this progression as a staged capability build-out MIL and SIL capabilities extended first, since both can be developed largely independently of final hardware, followed by PIL and HIL capabilities timed to arrive alongside, rather than after, the centralized compute platform itself.

5. Discussion

5.1 Platform Transition Strategy: Greenfield SDV and Legacy Fleet Bridging

The transition to centralized or zonal compute is, in practice, two distinct programs rather than one, summarized in Table 2. A greenfield approach designs a new machine platform around centralized compute, an Ethernet backbone, and OTA-capable connectivity as baseline requirements from the program. inception, which is the cleanest technical path but requires the validation infrastructure investment described in Section 4.3 to be committed before the first production unit ships. A bridging approach instead layers telematics gateways and edge-compute retrofits onto the existing domain-architecture fleet, aggregating legacy CAN network data, exposing it over IP to cloud-based analytics, and accepting limited configuration updates over cellular or satellite links without attempting to make the underlying domain ECUs directly reprogrammable [24].

Dimension	Greenfield SDV Platform	Legacy Fleet Bridging
Capital investment	High new compute platform plus validation infrastructure	Lower telematics gateway / edge-compute retrofit
Connectivity scope	Full OTA-capable connectivity by design	Limited configuration updates and telemetry only
Achievable software-defined capability	Full: OTA updates, edge AI inference, continuous validation	Partial: telemetry aggregation, limited configuration updates
Typical fleet segment	New machine platforms	Machines with 10–15+ years remaining service life
Validation infrastructure impact	Requires new integration-capable HIL from program start	Minimal domain ECUs remain independently validated

Table 2. Platform-transition strategy comparison: greenfield software-defined design versus legacy-fleet bridging [6].

The two approaches are not competing options for the same fleet; they serve different populations of the same manufacturer's installed base, and the practical challenge is less architectural than organizational, sustaining clear separation of program objectives and engineering resource allocation between a greenfield centralized-platform program and a bridging program for machines already in the field, a separation that is straightforward to state on a roadmap and considerably harder to hold in practice once engineering capacity is under pressure. A common failure mode is allowing the bridging program's near-term deliverables to consume engineering resources originally allocated to the greenfield program's validation infrastructure build-out, on the reasoning that the bridging work is more urgent because it touches the existing revenue-generating fleet. This is an understandable prioritization in the short term, but it directly undermines the timeline for the capability the greenfield program actually depends on.

The wiring-harness cost data from automotive E/E architecture analysis is instructive here even though it does not transfer directly: harness costs account for roughly 20 percent of total E/E architecture budgets in a modern passenger vehicle [8], and off-highway harness runs are typically longer and more exposed than automotive equivalents, which means the harness-simplification benefit of centralization, while real, arrives well after the validation-infrastructure investment described above, not before it a sequencing point that matters for how a greenfield

program's costs should be staged and justified internally. Presenting the harness-cost savings as an early return on the centralization investment, without also presenting the validation-infrastructure cost that has to be paid first, risks setting an internal expectation the program cannot meet on the timeline implied [23].

The organizational failure mode described above is easiest to prevent with an explicit resource-allocation rule rather than a general intention to keep the two programs separate: a rule specifying, for instance, that bridging-program engineering requests above a defined threshold require sign-off from the greenfield program's own leadership, not just from the bridging program's. This kind of structural safeguard is unglamorous compared to the technical content of either program, but it is exactly the mechanism that keeps a stated roadmap priority from eroding under day-to-day schedule pressure [22].

5.2 Middleware Standardization and the Case for Shared Software Platforms

The absence of an off-highway equivalent to the AUTOSAR Adaptive Platform compounds the centralization challenge rather than causing it independently. Kugele, Obergfell, and Sax's model-based resource synthesis approach for service-oriented automotive architectures depends on a stable middleware target to design against [6]; without a comparable off-highway standard, each manufacturer and Tier 1 supplier builds proprietary middleware layers with incompatible service-discovery, communication, and diagnostic conventions, which means application software validated on one centralized platform cannot be ported to another without substantial rework. This is a real cost, but it is a cost that compounds the validation infrastructure burden rather than substituting for it: even a manufacturer operating within a single, internally standardized middleware stack still faces the HIL infrastructure replacement problem described in Section 4.2, because the compute-architecture centralization decision and the middleware standardization decision are separable, and off-highway programs that treat them as a single bundled problem risk underestimating how much of the difficulty is architectural rather than software-standards driven.

Wang et al.'s observation that domain, zonal, and centralized architectures should be evaluated in parallel rather than as a forced sequence applies with equal force to middleware strategy: an off-highway manufacturer does not need industry-wide middleware convergence to begin the centralization transition, though convergence would reduce its long-run cost by allowing application software and validated components to be shared across programs rather than redeveloped for each proprietary stack [1]. In the absence of an industry-wide standard, a manufacturer's most practical path is internal standardization, committing to a single middleware layer across its own product lines even before an industry-wide equivalent to AUTOSAR Adaptive exists for machinery, since internal consistency captures much of the software-reuse benefit that full industry convergence would eventually provide without requiring the manufacturer to wait for a standardization effort it does not control.

Internal standardization also has a validation-infrastructure benefit that compounds with the argument developed in Section 4.3: a single internal middleware stack means HIL, SIL, and PIL test harnesses built for one product line's centralized platform can be adapted, rather than rebuilt from scratch, for the next. This reuse does not eliminate the validation-infrastructure investment this article treats as the actual limiting factor, but it does reduce its marginal cost for each subsequent platform program after the first, which is a meaningful consideration for a multi-product-line manufacturer deciding how to sequence centralized-platform rollouts across its portfolio [26].

5.3 Functional Safety and Cybersecurity Implications of Centralization

Centralizing compute changes the functional safety and cybersecurity posture of an off-highway machine in ways that a domain architecture's physical ECU separation previously handled implicitly. In a domain architecture, a software fault is largely contained within its own ECU, and the attack surface exposed to external networks is limited by the number of ECUs with any external connectivity at all. On a centralized platform, a compromised or faulty component can potentially affect every software function sharing that compute resource, which requires the hypervisor- or operating-system-level isolation mechanisms discussed in Section 3.2 to be evaluated explicitly against IEC 62061 and ISO 13849's safety-integrity-level architecture requirements [4], not assumed from the platform's automotive ASIL certification if one exists, since the machinery standard's own requirements govern.

The cybersecurity implications compound this. SAE J3061's guidebook for cyber-physical vehicle system threat analysis and risk assessment methodology [21] and Mahmood, Nguyen, and Shaikh's systematic threat assessment of automotive OTA mechanisms [9] both establish that an OTA-capable, network-connected platform requires threat modeling that a disconnected or minimally connected domain architecture did not, accounting for the full update-delivery attack surface; establishing secure boot and cryptographic identity chains for every node capable of receiving software; and defining an incident-response process for a platform that may not be reachable for rapid remote intervention. Pascale et al.'s intrusion detection work for connected vehicles demonstrates that network-level monitoring is a necessary complement to update-channel hardening rather than an alternative to it [10], a point that applies directly to an off-highway centralized platform accepting cellular or satellite-delivered updates at a remote job site with limited opportunity for rapid incident response [21].

None of this cybersecurity and functional safety burden is unique to off-highway; what is specific to the domain is that it arrives simultaneously with the environmental qualification and service life constraints discussed in Section 3.2, on an engineering organization whose existing safety case was built around ISO 26262-adjacent or machinery-standard practice for physically isolated ECUs, not around a shared, centralized compute platform hosting mixed-criticality functions under a connectivity profile the organization has not previously had to defend [22]. Building this capability from a standing start, rather than incrementally extending an existing automotive-grade cybersecurity practice the way many automotive OEMs can, is itself part of the validation infrastructure and organizational capability burden this article treats as the actual limiting factor on off-highway centralization [23].

A further complication specific to off-highway equipment is that many machines are resold, leased, or transferred between operators over their long service lives, which means the cryptographic-identity and secure-boot chains established in the threat-modeling framework above must also account for a change-of-custody event that automotive OTA threat models, built around a comparatively shorter and more centrally tracked ownership chain, do not typically treat as a first-class scenario [24].

Limitations.

This analysis is a literature-grounded comparative framework rather than an empirical study of a specific fielded platform, and its centralization-threshold argument would benefit from validation against a documented off-highway centralization program's actual cost and schedule data, which is not yet available in the published literature reviewed here. The framework also does not quantify the relative cost of the greenfield and bridging strategies directly, since neither the automotive nor the off-highway literature currently reports comparable figures for machinery-specific programs [24]. Finally, the analysis draws its off-highway autonomy examples predominantly from mining and agricultural applications, where published literature is comparatively available; construction equipment, which faces its own distinct mix of environmental and regulatory constraints, is addressed only through the general off-highway constraint set in Section 3.2 rather than through dedicated construction-specific literature [25].

The article also does not address the human-factors and workforce-training dimension of the centralization transition: how field service technicians and equipment operators, trained on domain-architecture diagnostic tools and procedures, are retrained for a centralized platform's different failure modes and diagnostic interfaces, a dimension the literature reviewed here does not yet cover in the off-highway context and that would benefit from dedicated treatment [26].

6. Conclusions

The central claim of this analysis is that compute-architecture centralization, not middleware fragmentation or field connectivity, is the binding constraint on software-defined capability in off-highway equipment and that the constraint is tighter for off-highway than for automotive because the domain's own requirements extended environmental qualification, machinery-specific functional safety frameworks, long service lives, and a validation infrastructure built around function-isolated HIL practice raise the cost of the domain-to-centralized transition beyond what automotive's more mature supply chain absorbs. The threshold at which centralization becomes necessary is

reached not by any single autonomy function but by the combination of onboard AI inference, OTA-deliverable updates, and continuous validation operating together, and the resource an off-highway organization is least prepared to scale incrementally to meet that threshold is integration-capable validation infrastructure, not compute hardware.

For engineering leadership weighing this transition, the practical implication is a sequencing one: a platform-transition framework distinguishing greenfield software-defined design from legacy-fleet bridging offers a way to sequence the required investment, provided the harness-simplification and connectivity benefits of centralization are understood to arrive after the validation-infrastructure cost, not before it, and provided the validation-infrastructure and staffing implications are budgeted alongside the compute-hardware and middleware costs that dominate most current program planning. Treating the centralization decision as primarily a hardware and software-standards question, as the automotive-derived literature tends to frame it, risks committing an off-highway program to a platform it cannot fully qualify for field release on the timeline that program planning assumed. Future work applying this framework against a documented off-highway centralization program's actual cost and schedule data would test the centralization threshold argument directly, in a way the currently available literature does not yet permit.

This reframing carries a specific implication for how off-highway engineering organizations structure their own internal decision processes: the centralization decision should not be delegated solely to compute-hardware or software-architecture teams, since the criterion this analysis identifies as dominant validation-infrastructure scaling sits organizationally with validation and test engineering, a group that is not always represented at the point where architecture decisions are made. Involving that group early, with real budget authority rather than a consultative role only, is one of the more directly actionable implications of the argument developed here, independent of which specific platform-transition strategy a given organization ultimately chooses.

The broader pattern this analysis surfaces, a technically mature literature developed in one domain being applied to an adjacent domain without examining whether its central assumptions transfer, is not unique to software-defined vehicles, and the same discipline of checking domain-specific constraints before adopting an adjacent field's framing would likely repay attention in other areas where off-highway engineering borrows heavily from automotive practices.

Author Contributions:

Conceptualization, Methodology, Investigation, Writing – Original Draft, Writing – Review & Editing, Visualization: S.C. The author has read and agreed to the published version of the manuscript.

Funding:

This research received no external funding.

Institutional Review Board Statement:

Not applicable. This study did not involve human participants, animal subjects, or identifiable personal data requiring ethics board review.

Informed Consent Statement:

Not applicable.

Data Availability Statement:

No new data were generated in this study. All analysis is based on the cited literature and publicly available standards documentation.

Conflicts of Interest:

The author declares no conflict of interest.

REFERENCES

1. W. Wang, K. Guo, W. Cao, H. Zhu, J. Nan, and L. Yu, "Review of Electrical and Electronic Architectures for Autonomous Vehicles: Topologies, Networking and Simulators," Automotive Innovation, vol. 7, no. 1, pp. 82–101, 2024.
2. AUTOSAR Consortium, AUTOSAR Adaptive Platform Specification R22-11. Available online: https://www.autosar.org (accessed on 18 August 2026).
3. L. Mauser and S. Wagner, "Centralization potential of automotive E/E architectures," Journal of Systems and Software, vol. 219, art. 112220, 2025.
4. ISO 13849-1:2023 / IEC 62061:2021, Safety of Machinery: Safety-Related Parts of Control Systems. Available online: https://www.iso.org (accessed on 18 August 2026).
5. M. Broy, I. Kruger, A. Pretschner, and C. Salzmann, "Engineering Automotive Software," Proceedings of the IEEE, vol. 95, no. 2, pp. 356–373, 2007.
6. S. Kugele, P. Obergfell, and E. Sax, "Model-based resource analysis and synthesis of service-oriented automotive software architectures," Software and Systems Modeling, vol. 20, no. 6, pp. 1945–1975, 2021.
7. ISO 26262:2018, Road Vehicles: Functional Safety. Available online: https://www.iso.org (accessed on 18 August 2026).
8. McKinsey & Company, "Getting Ready for Next-Generation E/E Architecture with Zonal Compute," 2023. Available online: https://www.mckinsey.com/industries/semiconductors/our-insights/getting-ready-for-next-generation-ee-architecture-with-zonal-compute (accessed on 18 August 2026).
9. S. Mahmood, H. N. Nguyen, and S. A. Shaikh, "Systematic threat assessment and security testing of automotive over-the-air (OTA) updates," Vehicular Communications, vol. 35, art. 100468, 2022.
10. F. Pascale, E. A. Adinolfi, S. Coppola, and E. Santonicola, "Cybersecurity in Automotive: An Intrusion Detection System in Connected Vehicles," Electronics, vol. 10, no. 15, art. 1765, 2021.
11. M. Abboush, C. Knieke, and A. Rausch, "Advancing real-time validation of automotive software systems via continuous integration and intelligent failure analysis," Scientific Reports, vol. 15, art. 32936, 2025.
12. L. Mauser, E. Zimmermann, P. Nedvědický, T. Eisenreich, M. Wäschle, and S. Wagner, "Towards Mixed-Criticality Software Architectures for Centralized HPC Platforms in Software-Defined Vehicles: A Systematic Literature Review," in Software Architecture (ECSA 2025), Springer, Cham, 2026, pp. 144–160.
13. ISO/PAS 8800:2024, Road Vehicles: Safety and Artificial Intelligence. Available online: https://www.iso.org (accessed on 18 August 2026).
14. L. Chen, Y. Xie, Y. He, Y. Ai, B. Tian, L. Li, S. Ge, and F.-Y. Wang, "Autonomous mining through cooperative driving and operations enabled by parallel intelligence," Communications Engineering, vol. 3, art. 75, 2024.
15. E. Aydemir and M. Unel, "Motion Planning and Path Following for Autonomous Navigation and Reversing of a Full-Scale Mining Truck and Trailer System," International Journal of Automotive Technology, vol. 26, no. 3, pp. 595–606, 2025.
16. E. Essien and S. Frimpong, "Enhancing Autonomous Truck Navigation in Underground Mines: A Review of 3D Object Detection Systems, Challenges, and Future Trends," Drones, vol. 9, no. 6, art. 433, 2025.
17. H. Wu, X. Wang, X. Chen, Y. Zhang, and Y. Zhang, "Review on Key Technologies for Autonomous Navigation in Field Agricultural Machinery," Agriculture, vol. 15, no. 12, art. 1297, 2025.
18. S. Y. Liang, H. Wu, L. Zhen, Q. Z. Hua, S. Garg, G. Kaddoum, M. M. Hassan, and K. P. Yu, "Edge YOLO: Real-Time Intelligent Object Detection System Based on Edge-Cloud Cooperation in Autonomous Vehicles," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 12, pp. 25345–25360, 2022.
19. J. Nibert, M. E. Herniter, and Z. Chambers, "Model-Based System Design for MIL, SIL, and HIL," World Electric Vehicle Journal, vol. 5, no. 4, pp. 1121–1130, 2012.
20. J. Khan and C. S. Vijay, "Validation of Safety of the Intended Functionality for Autonomous Driving Systems," presented at the 27th International Technical Conference on the Enhanced Safety of Vehicles (ESV), Yokohama, Japan, 2023, Paper 23-0009.
21. SAE J3061_202112, Cybersecurity Guidebook for Cyber-Physical Vehicle Systems. Available online: https://www.sae.org (accessed on 18 August 2026).
22. J. Ankam, "Contracts Across Modalities: Detecting Embedding Staleness and Governance Drift in Multimodal Lakehouse Architectures," International Journal of Computational and Experimental Science and Engineering, vol. 10, no. 1, 2024. Available online: https://doi.org/10.22399/ijcesen.5469.
23. J. Ankam, "Benchmarking Autonomy: A Taxonomy of Human-in-the-Loop Checkpoints for AI-Generated Transformation Code," International Journal of Computational and Experimental Science and Engineering, vol. 8, no. 3, 2022. Available online: https://doi.org/10.22399/ijcesen.5462.
24. S. K. Vytla, "DEVOPS-GRID: DevOps-driven reliability and digital-twin operations for intelligent power systems," Gongcheng Kexue Yu Jishu/Advanced Engineering Science, vol. 57, no. 3, 2025.
25. S. K. Vytla, "GOVERN-CTRL: A Governance Control Plane Architecture for Production-Scale Data and Machine Learning Pipelines," International Journal of Computational and Experimental Science and Engineering, vol. 12, no. 3, 2026. Available online: https://doi.org/10.22399/ijcesen.5394.
26. S. K. Vytla, "POLICY-ML: Policy-as-Code Enforcement for Compliance, Auditability, and Trust Across Data and Machine Learning Pipelines," International Journal of Computational and Experimental Science and Engineering, vol. 10, no. 4, 2024. Available online: https://doi.org/10.22399/ijcesen.5393.

27. H. Gaggar, "Architecting Scalable Enterprise AI Agent Infrastructure for Distributed Intelligent Systems," International Journal of Advances in Signal and Image Sciences, pp. 193–208, 2025. Available online: https://doi.org/10.29284/da9w0117.