

A platform-evidence mapping for DAL-B certification planning on an AMD64 multi-core processor with Xen temporal isolation

Micheal Saleab

`micheal.saleab@amd.com`

Advanced Micro Devices (Germany)

Stefano Stabellini

Advanced Micro Devices (United States)

Andrea Bastoni

Technical University of Munich

Giulio Corradi

Advanced Micro Devices (Germany)

Florian Holzapfel

Technical University of Munich

Kirk Baugher

Advanced Micro Devices (United States)

Research Article

Keywords: AMD64, DO-178C, AC 20-193, DAL-B, Cache coloring

Posted Date: October 22nd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11271040/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

A platform-evidence mapping for DAL-B certification planning on an AMD64 multi-core processor with Xen temporal isolation

Micheal Saleab^{1,4*}, Stefano Stabellini², Andrea Bastoni³,
Giulio Corradi¹, Florian Holzapfel⁴, Kirk Baugher²

^{1*}Advanced Micro Devices GmbH, Willy-Brandt-Allee 4, München, 81829, Germany.

²Advanced Micro Devices, Inc., 2100 Logic Drive, San Jose, CA, 95124, USA.

³Chair of Cyber-Physical Systems in Production Engineering, Technical University of Munich, Boltzmannstr. 15, Garching b. München, 85748, Germany.

⁴Institute of Flight System Dynamics, Technical University of Munich, Boltzmannstr. 15, Garching b. München, 85748, Germany.

*Corresponding author(s). E-mail(s): micheal.saleab@amd.com;
Contributing authors: stefano.stabellini@amd.com;
andrea.bastoni@tum.de; giulio.corradi@amd.com;
florian.holzapfel@tum.de; Kirk.Baugher@amd.com;

Abstract

Civil-avionics multi-core integration is regulated by DO-178C and FAA Advisory Circular AC 20-193 with its EASA counterpart AMC 20-193, all of which require the applicant to identify and bound multi-core shared-resource interference. On the AMD64 reference configuration considered here, the shared last-level cache (LLC) is the interference channel analyzed in detail, and hardware way-partitioning alone does not bound the hypervisor's own LLC footprint. The assumed platform is a Type-1 static-partitioning Xen configuration on AMD64 that uses Xen temporal isolation together with LLC cache coloring across guests *and* the hypervisor itself. Combined with the publicly-documented Xen safety-certification program as hypervisor-element evidence and a lean, model-based DO-178C/DO-331 workflow for hosted applications, the argument provides a candidate platform-evidence structure for a DAL-B applicant program, subject

to authority concurrence, program-specific timing closure, COTS hardware assurance, and hosted-application DO-178C evidence. We give objective-by-objective coverage for AC / AMC 20-193 and the multi-core-touched subset of DO-178C Annex A; identify DAL-A gaps; and ground the argument in a concrete UAS partition configuration. The certification mapping presented here is one input to an applicant program’s certification package; authority concurrence and hosted-application DO-178C evidence remain program responsibilities.

Keywords: AMD64, DO-178C, AC 20-193, DAL-B, Cache coloring

1 Introduction

Civil-avionics programs increasingly consolidate mixed-criticality workloads on multi-core system-on-chips: integrated modular avionics (IMA) partitions of different assurance levels, UAS autopilots that combine control with perception and mission computing, and UAM vehicles that place flight control beside general-purpose compute. The available compute platforms now include AMD64 alongside prior-generation embedded ISAs. DO-178C [22], FAA AC 20-193 [6] and EASA AMC 20-193 [2] require two platform preconditions for this setting: bounded multi-core interference and a platform-level safety case for the hypervisor element.

For the reference configuration considered here, both preconditions are tied to public technical artifacts. The assumed temporal-isolation platform is a Type-1 static-partitioning Xen hypervisor on AMD64 with LLC cache coloring enabled (Section 3): the LLC is partitioned at page granularity, each guest is assigned a disjoint color set, and the hypervisor itself is bound inside a disjoint color set — its text, data and order-zero heap relocated at boot so that hypervisor execution does not evict guest cache lines. These platform properties are treated as configuration assumptions for the certification mapping.

The second precondition — a safety case for the hypervisor element — may be supported by reusing element evidence produced for the hypervisor independently of the applicant’s own DO-178C effort, so the two evidence sources must be kept distinct. The airborne software in this paper is assessed against the DO-178C design assurance levels (DAL-A to DAL-E). Separately, the Xen hypervisor element is the subject of a publicly-documented safety-certification program [28, 33] that is producing a hypervisor-element artifact set — software safety requirements, software architecture, a hazard and operability analysis (HAZOP), a failure modes, effects and diagnostic analysis (FMEDA) and an element safety manual (ESM) — which an avionics applicant may reuse as element evidence for the hypervisor. The Xen Project marks its safety requirements as experimental and maintained on a best-effort basis [34]. This element evidence is distinct from, and not a substitute for, the DO-178C objectives the applicant owns; Section 6.1 sets out how it enters the DAL-B argument.

1.1 Certification-evidence reuse within the P100 Series

Certification-evidence reuse is governed by FAA AC 20-148 [4]: artifacts produced for a specific processor can be reused for subsequent applications on the same processor. Subject to the per-program impact analysis required by AC 20-148, an applicant could propose reusing the certification mapping on AMD Ryzen AI Embedded P100 Series processors (Section 3), provided that the same firmware options are used.

Contributions.

The paper makes three certification-methodology contributions.

1. It gives objective-by-objective coverage tables against FAA AC 20-193 with its EASA counterpart AMC 20-193 (Section 4), and the multi-core-touched subset of DO-178C Annex A (Section 5) for an AMD64/Xen temporal-isolation platform using LLC cache coloring.
2. It outlines a DAL-B applicability argument for the reference configuration (Section 6.1) and a concrete gap analysis for DAL-A (Section 6.2).
3. It extends the lean DO-178C/DO-331 model-based software development workflow of Dmitriev et al. [1] to AMD64 multi-core platforms (Section 7) and presents a concrete partition-configuration example.

The validity of the certification argument does not rely on any specific measured timing value (Section 6.1): the application-layer timing analysis of Saleab et al. [26] supplies the quantitative timing evidence, while the present article supplies the objective-level certification mapping for one specified AMD64/Xen configuration.

Scope and non-claims.

For the avoidance of doubt, the claim is narrow: the AMD64/Xen temporal-isolation platform described in Section 3, together with the publicly-documented Xen safety-certification program [28], defines candidate platform inputs for evaluating one AMD64 configuration under the robust-partitioning verification path of AC 20-193 §5.5. Tables 2 and 3 map those inputs to the relevant objectives; Table 4 states what a DAL-A claim would still require; Table 5 separates platform, applicant and authority responsibilities; and Section 7 shows where the three platform hooks enter the lean DO-178C/DO-331 workflow. Figure 1 summarizes this storyline.

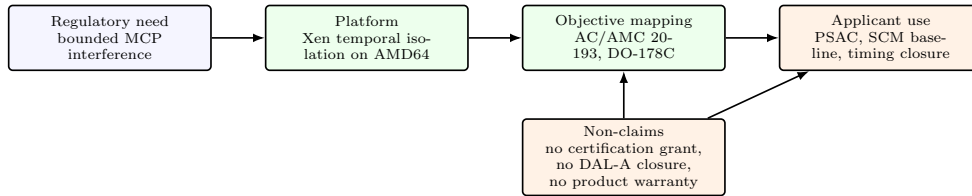


Fig. 1 Self-contained argument map. The article first defines the certification need, then states the AMD64/Xen platform, maps that platform to the relevant multi-core objectives, and separates applicant-owned evidence and non-claims. The diagram is methodological and does not assert certification approval for any aircraft system or product.

We do *not* claim:

- that any aircraft system is hereby certified;
- that an applicant program has obtained DER or CVE concurrence on the present argument;
- that the DAL-A gap is closed by this article;
- that AI / ML workloads on the integrated graphics unit are treatable today (Section 8, F4);
- that cache coloring is the only mechanism that can support a DAL-B argument on AMD64 — Table 1 (Section 3.1) enumerates the alternatives;
- that a Type-1 hypervisor, or Xen specifically, is the only way to obtain robust partitioning on AMD64 — bare-metal deployments and other hypervisors can also use hardware way-partitioning, as Table 1 records; or
- that the hosted-application DO-178C evidence is contributed by the platform (it remains the applicant’s responsibility at any DAL).

This article is a certification-methodology contribution; four of the authors are employed by AMD, and the article is written in their personal academic capacity, not as an independent product assessment. No statement herein is a representation or warranty by AMD or any silicon vendor that an AMD64 product is fit for safety-critical airborne use. Consistent with industry practice for COTS multi-core processors, the applicant program addresses the silicon in accordance with the COTS-component usage guidance of DO-254 §11.2 (and §11.3 where applicable) together with AC 20-152A §6 (Section 6.2, gap G5).

2 Regulatory and literature context

2.1 DO-178C timing evidence and the multi-core gap

DO-178C [22] defines the software design assurance levels and the Annex A objectives used by civil-avionics applicants; DAL-A programs satisfy all 71 objectives, DAL-B 69 and DAL-C 62. Its timing objective assumes that executable object code can be shown to meet its high-level timing requirement. On a single-core target, a task’s worst-case execution time (WCET) is bounded in isolation and worst-case response time (WCRT) adds the scheduler and interrupt terms. On a multi-core target, that argument requires shared-resource isolation: co-resident software can inflate the memory-access penalty through the shared LLC, DRAM, TLB, interconnect and interrupt paths. That this interference is a first-order threat to multi-core timing predictability is well established in the real-time and certification literature [8, 18]. The application-layer studies of Saleab et al. [25, 26] quantify this effect specifically on UAS flight code and map it to DO-178C timing evidence.

The reference configuration of Section 3 therefore separates the WCRT input into three platform terms. Cache coloring bounds the LLC contribution by assigning each partition and the hypervisor itself a disjoint color set. The Xen `null` scheduler removes preemption between hosted partitions in the reference configuration. IRQ-delivery latency is treated as an empirical term that the applicant bounds for its own platform and stress model. For DAL-A and DAL-B flight compute, periodic tasks released at

typical 50–1000 Hz flight-control-loop rates [29] must complete within their deadlines on every iteration. The platform contribution is not the applicant’s schedulability analysis itself; it is the isolation mechanism and the WCET/WCRT input structure on which that analysis can rely.

2.2 AC / AMC 20-193 and COTS hardware

FAA AC 20-193 [6] and EASA AMC 20-193 [2] express the same certification need in multi-core-processor (MCP) terms: the applicant must identify shared resources, define their intended use, mitigate or bound interference and report the resulting compliance argument. Section 4 presents a single consolidated MCP-coverage table for them. Earlier cache-partitioning certification work [12] establishes the same pattern on prior-generation embedded MCPs; the certification mapping below applies that pattern to the AMD64 Type-1 hypervisor case. The static-partitioning Xen configuration is inside the AC 20-193 envelope because the `null` scheduler binds each virtual CPU (vCPU) to a fixed physical CPU (pCPU); dynamically allocated software applications are out of scope under AC 20-193 §2.6.4.

The applicability of DO-254 [21] is predominantly for custom and configurable airborne electronic hardware, not to commercial-off-the-shelf processors developed outside an avionics assurance process; however, applicants use DO-254 §11.2 COTS-component usage and possibly §11.3 service-experience for COTS CPUs. AC 20-152A §6 [5] further supplements DO-254 §11.2 with specific COTS objectives. The argument keeps the AMD64 silicon on that COTS path and focuses on the Type-1 hypervisor interference mechanism.

2.3 Related Work and positioning

The closest certification-oriented line is the Single Core Equivalent work of Pellizzoni, Mancuso, Caccamo and Sha [8, 18], which combines predictable execution, cache management, DRAM-bank partitioning and bandwidth regulation so that a task can be analyzed as if it ran on a dedicated core. OS-level mechanisms such as Colored Lock-down [15], PALLOC [37], MemGuard [36], MemPol [38] and X-Stream [31] supply complementary cache and memory-bandwidth controls.

Static-partitioning hypervisors brought the same idea closer to IMA deployments. Prior hypervisor-cache-coloring work on embedded-class [13, 16, 17] and on commodity 64-bit ISAs [10, 11, 14, 27] shows that page coloring is feasible, but those systems color guest memory or expose a virtual LLC; they do not include the hypervisor’s own text, data and order-zero heap inside the partitioning boundary. The present argument assumes this hypervisor-self-colored property as a configured feature of the AMD64/Xen reference platform and maps its certification consequences.

Commercial IMA platforms with public-record multi-core certification artifacts [9, 30, 32] already pair a hypervisor-element certification artifact pack with ARINC 653-style partition attributes. The AMD64/Xen temporal-isolation platform considered here occupies the same architectural slot and uses software page-granularity coloring to make the hypervisor itself part of the LLC-isolation discipline. The lean DO-178C

/ DO-331 workflow of Dmitriev et al. [1] provides the hosted-application process that Section 7 extends with three platform hooks.

3 Reference platform and temporal-isolation discipline

Reference platform.

The reference platform is a Type-1 hypervisor (Xen) on an AMD Ryzen AI Embedded P100 Series processor, configured for static partitioning. The approach can be scaled to the AMD Ryzen AI Embedded X100 Series, but it has not been evaluated on that series in this paper. The Xen null scheduler binds each virtual CPU to a fixed physical CPU. Partitioning is established at boot by the Hyperlaunch framework, which constructs each domain directly from a static boot description — the same boot-time domain-construction model as Arm *dom0less* — so that a DAL-relevant DomU is built and isolated at boot independently of Dom0, rather than being created and managed by Dom0 after the fact. I/O devices are statically passed through to the owning partition, so shared-I/O contention between partitions is removed by construction rather than bounded statistically. Guests run on top: Linux serves as Dom0, and a real-time operating system serves as the DAL-relevant DomU.

Xen LLC cache-coloring discipline.

Each physical page belongs to an LLC *color* determined by the address bits that select its cache set; pages of distinct colors cannot evict one another at the LLC. The coloring discipline assigns each guest a disjoint color set declared as a partition attribute, and crucially applies the same discipline to the hypervisor itself: its text, its data and its order-zero heap are allocated only from a hypervisor-only color set, and the hypervisor binary is relocated at boot onto a color-disjoint sequence of physical frames. Three engineering steps are required. (i) *Cache-aware page allocator*: every page handed to a guest or to the hypervisor’s heap is drawn from the requesting principal’s color set; the allocator maintains a per-color free-list. (ii) *Colored relocation of the hypervisor binary*: at early boot, before any guest is started, the Xen image is copied to a non-contiguous set of physical frames whose colors all lie inside Xen’s color set, and the page tables and trampoline are rebuilt accordingly. (iii) *Layout-aware virtual-to-machine address translation*: the standard `virt_to_maddr` path is extended to walk the color-set-aware machine frame number (MFN) sequence inside the Xen image range during early boot.

Firmware prerequisite.

Classical cache coloring assumes a linear physical-address-to-set mapping. Modern LLCs commonly distribute physical addresses across cache slices through a hardware hash, which breaks that assumption [27, 35]. The coloring discipline used here therefore carries a platform precondition: the LLC address-to-set mapping must be deterministic for the chosen silicon and firmware so that the per-partition color-disjointness invariant holds. On the AMD Ryzen AI Embedded P100 Series configuration considered here, disabling the LLC hash in firmware restores the linear address-to-color mapping used

by the coloring allocator: the color is derived directly from the relevant physical-address bits rather than from a hashed slice-selection function. The applicant validates the actual address-to-set behavior of its specific silicon revision and firmware version, configures the platform so that the mapping is deterministic, and freezes the resulting firmware settings in the configuration baseline.

Role split (platform / applicant / authority).

Stated in the three roles of Table 5, the temporal-isolation discipline of this section divides as follows.

- **Platform supplies:** the Type-1 static-partitioning configuration (null-scheduler vCPU pinning, Hyperlaunch static domains, static I/O passthrough) and the hypervisor-inclusive LLC cache-coloring mechanism, including colored relocation of the Xen binary onto a color-disjoint frame sequence.
- **Applicant must produce:** validation of the address-to-set behavior for the specific silicon revision and firmware version, per-partition timing measured on the target part, and the frozen configuration baseline that records these settings.
- **Authority must concur with:** the configuration baseline and the platform-boundary claim as part of PSAC review at SOI #1.

Platform boundary.

The certification argument uses only the following configured platform properties: disjoint color sets per principal, hypervisor self-coloring, firmware-deterministic set mapping and applicant-supplied timing terms summarized in Table 8. Figure 2 shows the evidence boundary used by the rest of the argument.

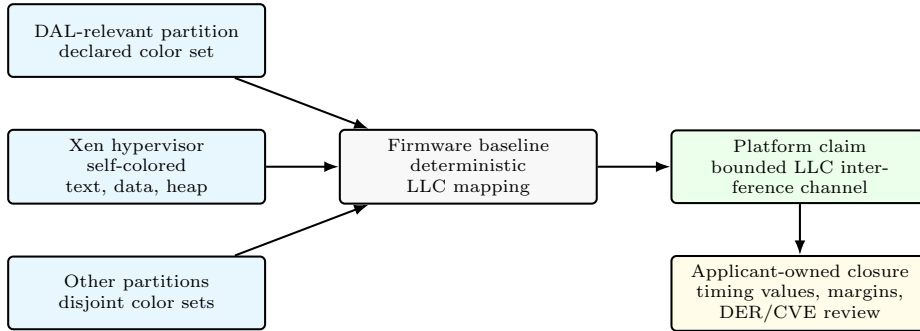


Fig. 2 Platform evidence boundary. The certification argument relies on the configured Xen temporal-isolation platform: LLC cache coloring, hypervisor self-coloring and firmware-deterministic LLC mapping. Program-specific timing values, margins and acceptance criteria remain applicant-owned evidence.

3.1 Cache coloring as an enabling isolation mechanism for a DAL-B argument on AMD64

A natural certification question is whether cache coloring is the only path to a DAL-B argument on AMD64. The answer is no: the certification need is shared-resource isolation, and Xen LLC cache coloring is the AMD64 mechanism considered here. This subsection states the precise claim that the rest of the argument relies on:

Cache coloring is not the only path to a DAL-B compliance argument on AMD64, but it is the path considered here because it simultaneously (i) supports evaluation under the robust-partitioning verification path of AC 20-193 §5.5 [6], (ii) bounds the hypervisor’s own contribution to the LLC interference channel constructively rather than statistically, (iii) scales the partition count beyond the small number of hardware ways, and (iv) preserves the multi-core consolidation assumed by the reference configuration. The alternatives considered below trade one of these four properties away.

The AC 20-193 §5.5 verification-path fork.

AC 20-193 MCP_Software_1 (§5.5) defines two verification paths. On an *MCP platform with robust partitioning*, applicants may verify each hosted software component *separately* on the MCP and determine its WCET separately. On *all other MCP platforms*, applicants must verify components that are not robustly partitioned by testing on the target MCP *with all hosted software executing in the intended final configuration*, including robustness testing of the MCP interfaces. Separate per-partition verification is therefore available only when the platform can justify robust partitioning. Cache coloring is the mechanism that provides the LLC-layer contribution to robust resource partitioning and robust time partitioning for the AMD64 reference platform.

Why the robust-partitioning path needs the hypervisor inside the discipline.

The *robust partitioning* definition of AC 20-193 §4 requires that one software partition cannot consume more than its allocation of any shared resource. The hypervisor is itself a software component; when it executes during VM exits, interrupt delivery, page-table walks or inter-partition messaging, its memory accesses contribute to LLC state. A mechanism that bounds guest LLC use but leaves the hypervisor unbounded therefore does not establish robust partitioning at the hypervisor layer; it merely reduces interference between guests. A constructive robust-partitioning argument at the hypervisor layer requires the hypervisor itself to operate inside a disjoint cache partition — the *hypervisor-self-colored* property described in Section 3. Without this property, robust partitioning is not established at the hypervisor layer and the applicant falls back to the all-together verification path.

High core-count scaling.

Without an LLC-isolation mechanism every additional active core becomes an additional potential attacker on every other core’s WCET, so the WCET budget of a hard-real-time partition grows with the active-core count and the incremental cores cease to be usable for hard-real-time work. Hardware way-partitioning caps the number of mutually-isolated partitions per LLC at the number of cache ways W (typically

4–16). Cache coloring lifts that cap to $N = C/(W \cdot \text{PAGE_SIZE})$, where N is the number of LLC-isolated partitions, C the LLC capacity, W the number of cache ways and PAGE_SIZE the page size (4 KiB here); this evaluates to 128 on an 8 MiB CCX and 256 on a 16 MiB CCX. The practical consequence is that a 16-core part can host more than W hard-real-time partitions only under a coloring discipline; under way-partitioning alone, additional cores do not by themselves create additional LLC-isolated partitions.

Composition with hardware way-partitioning.

Cache coloring and hardware way-partitioning operate on independent address-bit ranges and can be layered: the intersection of a Class of Service and a color set is well-defined. For the DAL-B argument below, coloring provides the LLC-layer evidence used for the AC 20-193 §5.5 robust-partitioning mapping; adding way-partitioning increases verification effort without changing that mapping. For a DAL-A claim, the combination yields two independent isolation mechanisms covering the same channel and is an optional credit item alongside the gaps cataloged in Section 6.2. AC 20-193 does not require multiple mitigations per channel; the combination is an engineering choice, not an obligation.

Alternative paths to a DAL-B applicability argument on AMD64.

Table 1 summarizes the design space of mechanisms that, in principle, can support a DAL-B applicability argument on AMD64 silicon. Each row describes one mechanism, the property it preserves and the property it sacrifices. Xen LLC cache coloring is the row in this comparison that preserves all four properties of the precise claim above. The hypervisor’s own contribution to the LLC channel is not a separate column: bounding it constructively is exactly what the *Hyp. RP* (hypervisor-layer robust partitioning) column records.

4 MCP objective coverage

AC 20-193 [6] and AMC 20-193 [2] share the same MCP-objective set (Section 2.2). This section maps each objective to the evidence produced by the platform of Section 3; the EASA reorganization is treated as equivalent in industry practice [20].

Applicability envelope.

Before the objective-by-objective mapping, we state the applicability envelope that the applicant’s PSAC must cite.

- **§2.2 — IDAL threshold.** AC 20-193 applies to MCPs with two or more activated cores whose hosted software or containing hardware item has IDAL A, B or C. The target of this paper is DAL-B (Section 6.1); the AC applies.
- **§2.6.4.1 — dynamic allocation excluded.** The Xen null scheduler binds each vCPU to a fixed pCPU for the life of the VM. The platform is *not* excluded under this clause.
- **§2.6.4 — SMT excluded.** SMT is disabled in firmware.
- **§2.6.4 — heterogeneous processors out of scope.** The platform uses a homogeneous core complex per partition; heterogeneous accelerators sharing the LLC are

Table 1 Alternative paths to a DAL-B applicability argument on AMD64. *Hyp. RP* marks whether the mechanism establishes constructive robust partitioning at the hypervisor layer; *Partitions / LLC* is the order-of-magnitude maximum useful partition count per LLC; *Consolidation* indicates whether the mechanism preserves multi-core consolidation. Xen LLC cache coloring is the row in this comparison that satisfies all four properties.

Mechanism	Hyp. RP	Parts. / LLC	Con-sol.	Principal limitation
Xen LLC cache coloring	Yes	128–512	Yes	Requires a deterministic LLC address-to-set mapping; where the LLC uses non-linear slice hashing this must be established for the chosen silicon and firmware.
Hardware way-partitioning alone	No	4–16	Yes	Per-ISA hardware way-partitioning / cache-allocation interface [19]. Hypervisor evictions are not bounded; falls back to all-together verification at the hypervisor layer. May still close the robust-partitioning path for bare-metal Linux deployments without a hypervisor.
One VM per physical core complex (CCX; no LLC sharing)	Yes*	= CCX count	No	* vacuous (no LLC sharing). Limits partition count to the number of physical core complexes (2–8 on contemporary parts); defeats the consolidation case.
Single-core equivalent (SCE) [8, 18]	Yes	N/A	No	Only one core active per critical task; trades compute density for predictability; may be impractical for UAS and IMA deployments that require consolidation.
Periodic LLC flush at partition switches	Partial	N/A	Yes	High overhead (cold-cache restart per window); does not isolate intra-window hypervisor execution; requires a partition scheduler.
Heavy measurement-based timing with extreme safety margin	Stat. only	N/A	Yes	May be acceptable in some lower-assurance contexts; DAL-B reviews generally favor constructive arguments, and large margins can consume the WCET budget of hard real-time tasks.

addressed as future work; excluding them here is a scope choice, not a statement that the mechanism cannot work on heterogeneous parts.

Role split (platform / applicant / authority).

The objective-by-objective mapping that follows is read in three roles, consistent with Table 5 and the columns of Table 2.

- **Platform supplies:** for each MCP objective, the configured platform property or cited element artifact recorded in the *Platform evidence* column of Table 2.
- **Applicant must produce:** the program-specific measurements and PSAC/SAS statements recorded in the *Applicant role* column — the resource-usage figures, the per-partition tests, and the hosted-software evidence that the platform does not supply.

- **Authority must concur with:** the objective alignment as presented in the PSAC and confirmed at the SOI reviews.

The mapping reduces to three checks: the PSAC freezes the AMD64/Xen configuration and shared-resource plan; the cache-colored hypervisor provides the LLC-partitioning input for `MCP_Resource_Usage_3` and the separate-verification path of `MCP_Software_1`; and the applicant remains responsible for application-level tests, coupling evidence, safety-net behavior and the SAS / HAS narrative.

4.1 Applicability beyond the null scheduler: time-sliced ARINC 653 partitions

The robust-partitioning argument under `MCP_Software_1` uses a null-scheduler configuration because it removes the scheduling term of the worst-case response time and so isolates the LLC contribution the platform mitigates. A real-world ARINC 653 program instead time-slices a major frame across partitions, but the coloring argument still holds: each partition's pages and the hypervisor's own text, data and heap stay color-disjoint across context switches. Because each partition keeps a statically assigned, disjoint color set, the design does not re-color memory at a partition switch; the number of concurrently color-isolated partitions is therefore bounded by the N available colors of Section 3.1 (the worked example of Section 7.1 partitions its 128 colors into four disjoint sets). The only added term is non-LLC — the applicant bounds the partition scheduler's worst-case context-switch cost from its own timing tests and adds it to each partition's WCRT budget.

5 DO-178C Annex A subset coverage

The MCP objectives touch only a subset of DO-178C Annex A. Table 3 lists that subset and the platform contribution to each objective.

The two Annex A objectives that distinguish DAL-A from DAL-B— A-7 Obj. 5 (modified condition/decision coverage, MCDC) and A-7 Obj. 9 (verification of additional code that cannot be traced to source code)—are hosted-application source-code objectives. They are the application developer's responsibility; no platform mechanism can supply them. Data-coupling and control-coupling coverage (A-7 Obj. 8) is *not* a DAL-A-only objective: it is required at both DAL-A and DAL-B and is therefore part of the DAL-B baseline, not a DAL-A gap. Section 6.2 returns to this point in the gap analysis.

6 DAL applicability argument

The platform of Section 3 provides inputs for a DAL-B applicability argument; a DAL-A argument requires the additional evidence items enumerated in Section 6.2.

Table 2 AC 20-193 objective alignment with the platform. The leading MCP_ prefix is omitted in the objective column for layout. MCP_Resource_Usage_2 is reserved in AC 20-193 and is covered by AC 20-152A objective COTS-8.

Objective	Platform evidence	Applicant role	Primary ref.
Planning_1 (§5.2.1)	Identifies MCP family, active cores, MCP architecture, hosted software components, dynamic features, IMA-platform status, and robust-partitioning status. The platform fixes these via a named silicon family, the firmware configuration baseline, the Xen null-scheduler configuration, and the Hyperlaunch static configuration.	PSAC and SAS reference the configuration sheet; SCM baseline.	§3
Planning_2 (§5.2.2)	Plans use of MCP shared resources to avoid or mitigate contention. Platform evidence: hypervisor-self-colored LLC plan, dual allocator, deterministic set-mapping established for the platform's firmware, and identification of safety-net candidates (Xen fatal-error handler, boot-time color-allocation checks).	PSAC describes the allocation and verification approach for shared resources.	§3
Res._Usage_1 (§5.3)	MCP configuration settings determined and documented: SMT off, LLC-hash disabled, C-states / boost set per safety configuration, memory interleaving fixed.	PSAC fixes the configuration baseline.	§3
Res._Usage_3 (§5.4)	LLC identified as the interference channel addressed by the platform, together with DRAM, TLB and interconnect. The cache-coloring discipline is intended to mitigate the LLC channel by construction, subject to validation of the address-to-set mapping and allocator invariants; because the hypervisor and a small RTOS partition can be configured with bounded LLC-resident working sets, coloring may also reduce downstream DRAM traffic.	Applicant extends the list with application-specific resources.	§3
Res._Usage_4 (§5.4)	Available resources of the MCP and interconnect identified; allocation per partition declared in the color-allocation table; worst-case demand compared with the applicant-supplied bounds summarized in Table 8.	Applicant verifies application-level demand against allocation.	§7.1
Software.1 (§5.5)	Software-verification objectives met under the robust-partitioning category: separate per-partition WCET and functional verification justified by the color-disjoint set mapping plus the hypervisor-self-coloring property.	Applicant provides per-partition tests traced to high-level requirements.	§3; §3.1
Software.2 (§5.5)	Data-/control-coupling between partitions on different cores is exercised through the declared inter-partition channels (event channels, shared-memory regions).	Applicant verifies data / control coupling on the integrated MCP per Annex A Table A-7.	§5
Error Handling 1 (§5.6)	Failure modes within the MCP identified; Xen fatal-error handler plus boot-time color-allocation checks form the platform safety net; failures contained inside the equipment.	Applicant defines the system-level error-handling and safety-net design.	Xen Project safety program
Accomp. Summary 1 (§5.7)	Not a platform artifact. The applicant summarizes in the SAS / HAS how each AC 20-193 objective is met for the specific program.	Applicant program; cross-references rows above.	DO-178C §11.20.i

Table 3 DO-178C Annex A objectives touched by the platform.

Table	Obj.	Objective	Platform contribution
A-3	5	Design specifies hardware/software interface.	LLC is a named interface element in AC 20-193; cache-coloring turns the LLC into a configured per-partition attribute.
A-3	6	Algorithms specified in design data.	Color allocation is a design-data artifact; §7 proposes a table format and tool integration.
A-5	1	Executable code complies with HLR including timing.	Timing bounded per partition by LLC coloring plus complementary bandwidth regulation; IRQ latency bounded by null-scheduler.
A-7	1	Test procedures correct.	Color allocation under test matches allocation under flight (configuration frozen at compile time).
A-7	2	Test results correct.	Results reproducible under static assignment of cores and shared resources; §3 defines the configuration baseline and Table 8 defines the timing-closure template.

6.1 A DAL-B applicability argument

This section outlines how an applicant program seeking certification of civilian airborne software at software design assurance level B could organize a platform applicability argument, given a platform configured as a Xen Type-1 static-partitioning hypervisor with the hypervisor-level LLC cache-coloring discipline of Section 3 enabled, the publicly-documented Xen safety-certification program evidence [28] applied to the hypervisor element and the lean DO-178C/DO-331 development process of Dmitriev et al. [1] applied to hosted applications. Here, *applicability argument* denotes constructive support for an applicant’s certification case under the stated configuration. The argument divides into the three roles of Table 5.

- **Platform supplies:** the Xen Type-1 static-partitioning configuration with hypervisor-inclusive LLC cache-coloring (Section 3) and the publicly-documented Xen safety-certification element evidence [28] applied to the hypervisor element.
- **Applicant must produce:** the hosted-application DO-178C Annex A evidence at DAL-B, the per-partition WCET and WCRT closure measured on the specific silicon, and the system-level safety net described under `MCP_Error_Handling_1`.
- **Authority must concur with:** the element boundary and the use-case limitations declared in the hypervisor vendor’s element safety manual, the configuration baseline, and each SOI checkpoint; concurrence by a designated engineering representative or equivalent authority remains the gating step between the proposed argument and a program-specific certification basis (Section 6.3).

Three observations support this outline.

First, every platform objective enumerated in Sections 4 and 5 is mapped to cited evidence, with residual items assigned either to the applicant program or to authority concurrence.

Second, the two Annex A objectives that DAL-A requires and DAL-B does not—A-7 Obj. 5 (MCDC) and A-7 Obj. 9 (verification of additional code that cannot be traced to source code)—are hosted-application objectives, not platform ones, and a DAL-B claim therefore does not hinge on them.

Third, the publicly-documented Xen safety-certification program [28, 33] identifies public hypervisor-element artifacts that the applicant’s PSAC may cite. Under SAE Aerospace Recommended Practice (ARP4754A) [24], such hypervisor-element evidence may be cited as supporting evidence for the DAL-B platform-layer argument, subject to the PSAC documenting the element boundary, the use-case limitations declared by the hypervisor vendor’s element safety manual, and the residual hosted-application objectives that remain the applicant’s DO-178C responsibility.

Robustness to applicant-supplied timing values.

The DAL-B applicability argument above is structured around the configured platform properties and evidence roles: (i) the LLC interference channel is bounded constructively at the hypervisor layer, (ii) IRQ-delivery latency is an applicant-bounded empirical term represented explicitly in Table 8, and (iii) the underlying mechanism is expressible as a configured per-partition attribute that is verifiable by ordinary configuration-management means. Changes to the applicant-supplied timing values used with Table 8 would require rechecking only the program-specific WCRT closure; the objective-level mapping in Tables 2 and 3 depends on the mechanism and evidence roles defined here, not on externally reported numeric values.

6.2 DAL-A gap analysis

A DAL-A claim for the reference configuration requires the following evidence items that are outside the scope of the DAL-B applicability argument. Table 4 enumerates them.

G1 and G2 are application-level source-code objectives owned by the applicant’s development program. G3, G4 and G6 are program-level items that further work can supply. G5 is the standing COTS-processor item: for commercial-off-the-shelf processors, applicants normally use the COTS-component usage guidance of DO-254 §11.2 (and §11.3 where applicable) plus AC 20-152A §6.

6.3 Readiness check: who supplies what

A DAL-B applicability argument does not become certification evidence until an appropriately-empowered assessor concurs — a designated engineering representative (DER) in the FAA system, a certification verification engineer (CVE) in the EASA system, or an equivalent. The structured argument is staged by an applicant program for that concurrence; it is aimed at the UAS, eVTOL and urban-air-mobility programs whose deployment context motivated [1].

Three roles share the argument: the *platform* (a fixed external evidence package), the *applicant program* (the aircraft-system development organization), and the *authority* (DER, CVE, or equivalent). Table 5 enumerates the responsibility split row by row, with each row a single auditable evidence item traceable to a normative objective. Applicant programs may attach the platform evidence at three documents: the PSAC SCM baseline freezes the platform identification and configuration items (MCP_Planning_1, MCP_Resource_Usage_1 / _2); the SDP/SVP cite the configured color-allocation artifacts as the evidence supplying MCP_Resource_Usage_3

Table 4 DAL-A gap analysis. Gaps G1–G6 enumerate the evidence items required for a DAL-A claim but outside this DAL-B scope. The *Indicative effort* column reports purely illustrative order-of-magnitude FTE-month and calendar-month estimates; actual figures depend on the size of the hosted software, the maturity of the applicant’s tooling and the DER’s expectations. *Owner* identifies who carries the gap.

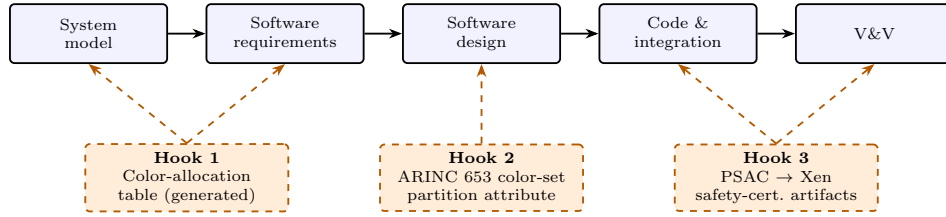
Gap	Required evidence	Indicative effort	Owner
G1	Modified condition/decision coverage (A-7 Obj. 5) on hosted application software.	Scales with hosted-application size; for a UAS-class autopilot typically 3–6 FTE-months on top of structural-coverage infrastructure.	Applicant
G2	Verification of additional code that cannot be traced to source code (A-7 Obj. 9; object-code-to-source traceability) on hosted application software.	Scales with the compiler / tool chain and the volume of non-traceable generated code; typically addressed by object-code verification or by using a qualified compiler.	Applicant
G3	DO-330 toolchain qualification: TQL-1 of the color-allocation generator (DAL-A jump from TQL-2 baseline) plus compiler and Xen build scripts.	6–12 calendar-months with third-party DO-330 support; ~2–4 FTE-months for the color-allocation generator alone once TQL-2 evidence (Section 7.2) is reused.	Program
G4	Formal verification of the colored-memory allocator, producing a machine-checked proof of the per-partition color-set disjointness invariant.	12–18 calendar-months with a formal-methods specialist; ~3–6 FTE-months core-mathematician effort plus support from a machine-checked proof assistant.	Program
G5	DO-254 evidence for the AMD64 silicon. The usual applicant path for COTS multi-core CPUs is the COTS-component usage guidance of DO-254 §11.2 (and §11.3 where applicable) together with AC 20-152A §6, as used by commercial IMA platforms.	Service-experience and COTS-evidence dossier assembled by the applicant program; effort scales with the depth of available field data and is typically a continuous activity rather than a fixed-duration item.	Applicant
G6	Independent third-party verification of the hypervisor element.	6 calendar-months at an ISO/IEC 17025-accredited laboratory; ~2–3 FTE-months program-side support.	Program

and `MCP.Software_1`; and the SAS records the applicant’s own `MCP.Software_2` and `MCP.Accomplishment.Summary_1` tests. A program that cannot populate the table row by row has not yet completed the platform portion of its SOI #1 preparation.

The table converts the headline claim into an auditable checklist: each *Platform or paper* row is an artifact or template the applicant may cite; each *Applicant builds* row is an evidence item the program owns; each *Authority concurs* row is a DER/CVE checkpoint. The table is a preparation aid for the platform portion of a DAL-B PSAC; it does not itself establish certification status for a program.

7 Extension of the lean DO-178C/DO-331 workflow

The lean workflow of Dmitriev et al. [1] was specified at the application layer and did not prescribe a compute platform. The workflow is adapted here to an AMD64/Xen temporal-isolation platform by identifying three hooks at which platform evidence plugs into the workflow’s artifact flow.



Solid: lean DO-178C / DO-331 workflow of Dmitriev et al. (2020); dashed: platform hooks added for the AMD64/Xen platform mapping.

Fig. 3 Extension of the lean DO-178C / DO-331 workflow to AMD64 multi-core. Three platform hooks plug the AMD64/Xen temporal-isolation platform into the model-based artifact flow: (1) the color-allocation table as tool-generated configuration data; (2) the color-set as a new ARINC 653 partition attribute; (3) the PSAC cross-reference to the Xen Project safety-certification program element artifact set.

Figure 3 shows the three hooks. Hook 1 adds the color-allocation table as tool-generated configuration data; the generator enforces color-set disjointness at build time. Hook 2 adds the color-set as an ARINC 653-style partition attribute alongside CPU affinity, memory footprint, device-passthrough set and scheduling budget. Hook 3 has the PSAC identify the hypervisor as a system element and reference the publicly-documented Xen safety-certification artifacts (Software Safety Requirements Specification, Software Architecture Specification, HAZOP, FMEDA and element safety manual) [28, 33].

Role split (platform / applicant / authority).

The three hooks connect platform-supplied evidence to artifacts the applicant owns and the authority reviews, in the roles of Table 5.

- **Platform supplies:** the generated color-allocation table (Hook 1), the partition-attribute declaration template (Hook 2), and the publicly-documented Xen safety-certification artifacts referenced at Hook 3.

Table 5 Evidence-allocation checklist for a DAL-B AMD64 IMA program using the AMD64/Xen temporal-isolation platform. Each row is a single evidence item. *Platform / mapping* marks evidence or templates provided by the platform description, the publicly-documented Xen safety-certification program, the lean DO-178C/DO-331 workflow or the certification mapping. *Applicant builds* marks evidence the applicant program produces (for the specific aircraft system, the specific application binary, the specific safety case). *Authority concurs* marks the gating step at which an appropriately empowered assessor reviews the evidence.

Evidence item	Platform or paper	Applicant builds	Authority concurs	Cross-reference
Cache-colored Xen build (mechanism + binary)	Yes	—	—	§3
Hypervisor-element safety case (HAZOP, FMEDA, ESM)	Yes	—	—	[28]
AC 20-193 objective alignment table	Yes	—	—	Table 2
DO-178C Annex A platform touch-points	Yes	—	—	Table 3
Lean DO-178C / DO-331 workflow	Yes	—	—	[1]
Worked partition-configuration template	Yes	—	—	Tables 6, 7
WCRT closure template	Yes	—	—	Table 8
DAL-A gap analysis with effort estimates	Yes	—	—	Table 4
PSAC scoping memo (program-specific)	—	Yes	—	§6.3
Program-specific configuration sheet (silicon part, firmware version, BIOS settings, Xen build tag, partition-attribute table)	—	Yes	—	§7.1
Hosted-application DO-178C evidence at applicable DAL (requirements, design, code, MCDC where required)	—	Yes	—	DO-178C Annex A
TOR/TR/TVCP for the color-allocation generator (or hand-authored allocation with independent review)	—	Yes	—	§7.2
Per-partition WCET and functional tests on the MCP	—	Yes	—	MCP_Software_1
Inter-partition data/control coupling tests	—	Yes	—	MCP_Software_2
System-level safety net response (alternate-channel handover, fail-operational design)	—	Yes	—	MCP_Error_Handling_1
Software accomplishment summary	—	Yes	—	AC 20-193 accomplishment-summary objective
COTS-component / service-experience dossier for the silicon	—	Yes	—	§2.2
PSAC concurrence at SOI #1	—	—	Yes	§6.3
Implementation review at SOI #2	—	17	Yes	DO-178C §10
Verification review at SOI #3	—	—	Yes	DO-178C §10
Final SAS concurrence at SOI #4	—	—	Yes	§6.3

- **Applicant must produce:** the populated and verified program-specific instance of each hook — the concrete color allocation, the concrete attribute table, and the PSAC text that cites the hypervisor element for the specific system.
- **Authority must concur with:** the resulting PSAC and verification artifacts at the corresponding SOI reviews.

7.1 Worked partition-configuration example

The worked example is a representative UAS-class IMA configuration with a DAL-B flight-control partition, a DAL-C perception partition and a DAL-D mission-management partition on an AMD64 four-core platform whose LLC exposes 128 available colors (Table 6). The example is a template: applicants substitute their own measured working sets, timing requirements and platform bounds into the same table layout.

Table 6 Worked partition-attribute declaration (Hook 2). Each row is one partition declared in the ARINC 653-style attribute table. *DAL* is the assurance level; *pCPU* is the dedicated physical core; *Timing policy* is the program-declared period/deadline class for the hosted task; *LLC working set* and *color share* are applicant-declared resource attributes recorded in the same configuration baseline.

Partition	DAL	Resource declaration	Role and sizing rationale
Flight control	B	Dedicated pCPU; applicant-defined timing policy; declared LLC working set and color share.	Inner-loop attitude control. The color budget exceeds the hottest measured working set from the cited flight-code timing study.
Perception	C	Dedicated pCPU set; applicant-defined timing policy; declared LLC working set and color share.	Vision/lidar processing on a dual-core partition. Half-cache allocation reduces DRAM traffic.
Mission	D	Dedicated pCPU; applicant-defined timing policy; declared LLC working set and color share.	Mission-management housekeeping. Sized symmetrically with the flight-control partition for configuration-audit simplicity.
Hypervisor (Xen)	—	Shared execution context; not periodic; declared LLC working set and reserved color share.	Xen self-coloring budget for text, data and order-zero heap; sized above the measured Xen working set.

Table 6 is the partition-attribute declaration produced at Hook 2 of Figure 3: one row per partition, with the color-set column added alongside the other ARINC 653 resource attributes that the applicant already maintains. Table 7 is the color-allocation artifact produced at Hook 1 by the workflow’s generator, derived deterministically from the declaration. The disjointness invariant is verified by the generator at build time; the derived allocation is frozen into the SCM baseline together with the Xen binary and its boot-time command line.

Table 7 Derived color-allocation table (Hook 1) produced by the workflow generator from the partition-attribute declaration. The table is a configuration-data artifact tracked in the SCM baseline and captured in the PSAC. Disjointness of color sets across partitions is enforced as a generator invariant and is independent of the order of partition declarations.

Partition	Color range	Disjointness check (generator invariant)
Xen (hypervisor)	0–27	disjoint vs. Flight/Perception/Mission
Flight control	28–52	disjoint vs. Xen/Perception/Mission
Perception	53–102	disjoint vs. Xen/Flight/Mission
Mission	103–127	disjoint vs. Xen/Flight/Perception

WCRT closure template for the DAL-B partition.

The configuration of Tables 6 and 7 provides the inputs used to structure the worst-case-response-time analysis of the DAL-B flight-control partition by composing the three component bounds of Section 2.1. Table 8 shows the closure form without declaring platform timing values: the applicant supplies the LLC-free execution term, the LLC-interference term, the IRQ-delivery term and any scheduling term for its own certified configuration. The aggregate WCRT is then compared with the applicant’s declared deadline and program-specific margin.

Table 8 Worst-case-response-time closure template for the DAL-B flight-control partition. C_{app} is the applicant-measured LLC-free execution term; Δ_{LLC} , Δ_{IRQ} and Δ_{sched} are applicant-bounded platform terms. The applicant supplies values, margins and acceptance criteria for its own configuration.

Term	Applicant entry	Evidence role
Application execution term C_{app}	measured by applicant	LLC-free or otherwise justified application timing baseline
LLC contribution Δ_{LLC}	bounded by applicant	residual cache-interference term for the configured color allocation
IRQ-delivery contribution Δ_{IRQ}	bounded by applicant	interrupt-delivery term under the program’s stress and load model
Scheduling contribution Δ_{sched}	bounded by applicant	absent only when the program justifies no partition preemption
WCRT relation	$C_{\text{app}} + \Delta_{\text{LLC}} + \Delta_{\text{IRQ}} + \Delta_{\text{sched}}$	additive bound to be justified in the timing-analysis report
Deadline D_{app}	declared by applicant	program-level timing requirement
Closure criterion	applicant-defined	WCRT must satisfy the declared deadline and DER-agreed margin

Representative PSAC excerpt.

At Hook 3 the applicant’s PSAC cites the Xen Project safety-certification program artifacts [28, 33] and the platform evidence rows of Tables 2–3. A representative PSAC excerpt reads as follows: “*The hypervisor element is the Xen build identified in the SCM baseline, configured with the cache-coloring discipline of Section 3 per the Xen command line and Hyperlaunch device-tree fragment recorded in Annex A of this PSAC (reproducing the partition-attribute and color-allocation declarations of*

Tables 6 and 7). Hypervisor-element evidence is cited from the publicly-documented Xen safety-certification program artifact set [28]. AC 20-193 platform evidence is traced in Annex B of this PSAC to the rows of Table 2 of the present paper. The hosted-application DO-178C evidence for the flight-control, perception and mission partitions is produced by the applicant program in accordance with the lean workflow of [1] and is the applicant’s responsibility at the corresponding DALs.”

The PSAC excerpt above is illustrative; the wording an applicant adopts is shaped by its own program template and by the DER’s preferences. The structural elements—hypervisor element identified, platform evidence cited, applicant responsibility split made explicit—are the intended contribution of the certification mapping.

7.2 DO-330 tool-qualification for the color-allocation generator

The Hook 1 color-allocation generator emits configuration data linked into the Xen image; a mis-computed color set would directly insert an error into the airborne software’s resource-isolation property. Under DO-178C §12.2.2 this places the generator in Criterion 1, which yields **TQL-2** for DAL-B hosted software via DO-178C Table 12-1 (and TQL-1 for DAL-A, listed as gap G3 in Section 6.2). TQL-2 requires the standard tool-qualification artifact set (TOR, TR, TAD, TVCP, TQP) per DO-330 [23]; the generator’s mechanically-checkable disjointness invariant keeps the test suite small. As defense in depth, Xen runs an independent boot-time invariant check that verifies (i) every DomU page belongs to its declared color set, (ii) DomU color sets are pairwise disjoint, and (iii) Xen’s own color set is disjoint from every DomU color set; failure is fatal at boot, so any tool error is contained before airborne software is exercised. An applicant may alternatively hand-author the color-allocation table and submit it as ordinary configuration data under DO-178C Annex A Table A-2 / A-3 (with independent disjointness review), in which case DO-330 qualification does not apply; the TQL-2 path is preferred when partition counts grow.

8 Conclusion and future work

We have outlined how an applicant program could organize a DAL-B applicability argument for the reference configuration, under a static-partitioning Xen configuration with the cache-coloring discipline of Section 3, the publicly-documented Xen safety-certification program [28, 33] hypervisor-element evidence, and the lean DO-178C / DO-331 workflow of Dmitriev et al. [1] for the hosted application. The argument is carried by the MCP coverage table (Table 2) and the DO-178C Annex A subset (Table 3). The DAL-A gap analysis (Table 4) identifies DO-330 toolchain qualification of the color-allocation generator (G3) and formal verification of the colored-memory allocator (G4) as the next items. Concurrence by a designated engineering representative or equivalent authority remains the gating step between the structured argument we present and any program-specific certification basis (Section 6.3).

Future work.

Five concrete directions follow. *(F1) DAL-A gap closure* — items G3, G4 and G6 of Table 4; G3 and G4 are research and engineering tasks, while G6 requires an independent laboratory engagement. *(F2) Composition with memory-bandwidth regulation* — MemGuard [36], MemPol [38] and PALLOC [37] address the orthogonal DRAM channel through bandwidth regulation and DRAM bank-aware allocation, and compose with cache coloring; extending the partitioning discipline to DRAM bank and channel granularity is a natural next step. Hardware quality-of-service (QoS) resource-partitioning interfaces offer a complementary hardware route whose real-time behavior has been evaluated empirically [19]. *(F3) Display-bearing IMA on AMD64 integrated-graphics platforms* — extending the coloring discipline to an integrated graphics unit that shares the LLC and interconnect (AC 20-193 §2.7). *(F4) Accelerator-bearing IMA under emerging AI guidance.* Cache coloring alone does not cover a future certification argument for AI / ML workloads on an integrated graphics unit: it bounds the CPU-iGPU shared resources, but learning assurance, AI explainability and the other elements of the EASA AI Concept Paper Issue 2 framework [3] and the FAA AI Safety Assurance Roadmap [7] remain open work outside the present DAL-B scope. *(F5) DER pilot program* — staging the argument with one DAL-B applicant program and one cognizant DER / CVE (Section 6.3) is a useful external validation item.

Threats to validity.

Four threats are explicit. *First*, the platform mechanism is evaluated only for the reference configuration described in Section 3; any material change to the mechanism, firmware mapping, or hypervisor configuration would require re-examining Tables 2–3. *Second*, no DER or CVE has yet concurred with the argument; the article provides a structured argument, not the certification basis (Section 6.3). *Third*, the platform evidence is stated for the specified AMD64 reference configuration only; reuse on other P100 Series processors requires the same firmware options and the impact analysis described in Sections 1.1 and 3. *Fourth*, the ARP4754A supporting-evidence framing used in Section 6.1 is a program-level claim that the applicant must evaluate against its own element-boundary analysis and the use-case limitations recorded in the hypervisor’s element safety manual; that element-boundary analysis is not performed in this article. These threats delimit the scope of the structural argument.

Author contributions. **M. Saleab:** Conceptualization, Methodology, Software, Investigation, Validation, Writing – Original Draft. **S. Stabellini:** Conceptualization, Methodology, Software, Investigation, Validation, Writing – Review & Editing. **A. Bastoni** and **G. Corradi:** Supervision, Conceptualization, Methodology, Writing – Review & Editing. **F. Holzapfel:** Supervision, Project administration, Writing – Review & Editing. **K. Baugher:** Writing – Review & Editing.

Data availability. This article is a certification-methodology article and does not report new measurement data. The platform evidence rows are framed as a self-contained certification mapping for the reference configuration. No sample-size-based empirical study is introduced here; sample-size annotation is therefore not applicable.

Funding. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. A. Bastoni was supported by the Chair for Cyber-Physical Systems in Production Engineering at TUM and the Alexander von Humboldt Foundation.

Declaration of competing interests. Authors M. Saleab, S. Stabellini, G. Corradi and K. Baugher are employed by Advanced Micro Devices, the silicon vendor whose AMD64 reference configuration is discussed in this article. Authors A. Bastoni and F. Holzapfel are employed by the Technical University of Munich, and M. Saleab is additionally a doctoral candidate at the same institution. A. Bastoni is also affiliated with Minerva Systems, which contributed the initial versions of cache coloring for Xen on Arm. The authors declare no other competing financial or personal interests.

Disclaimer. The AMD-affiliated authors contribute the argument presented in this article in their individual capacity as researchers; Advanced Micro Devices makes no representation or warranty, express or implied, regarding the fitness of any AMD64 silicon product for use in safety-critical airborne applications, and no statement in this article should be construed as such a representation, warranty or product commitment. The certification basis for any specific aircraft system remains the responsibility of the applicant program and the cognizant certification authority. The Xen Project is governed by the Linux Foundation; the hypervisor-element safety-certification artifacts referenced as publicly-documented in this article are produced by the Xen Project Safety Working Group and the wider Xen Project community, not by Advanced Micro Devices. All trademarks referenced are the property of their respective owners.

Acknowledgements. We thank the Xen Project community for its public safety-certification artifacts and the engineering teams for the platforms and firmware.

Abbreviations.

AC, AMC	Advisory Circular (FAA) / Acceptable Means of Compliance (EASA)
AEH	airborne electronic hardware (DO-254 / AC 20-152A)
ARINC 653	Avionics Application Software Standard Interface (APEX)
ARP4754A	SAE Aerospace Recommended Practice for civil-aircraft systems development
BIOS	basic input/output system (firmware)
CCX	core complex (AMD64 multi-core building block)
COTS	commercial off-the-shelf
CVE	certification verification engineer (EASA)
DAL, IDAL	(item) design / development assurance level (DO-178C, DO-254, ARP4754A)
DER	designated engineering representative (FAA)
DO-178C	RTCA Software Considerations in Airborne Systems and Equipment Certification
DO-254	RTCA Design Assurance Guidance for Airborne Electronic Hardware
DO-330	RTCA Software Tool Qualification Considerations
DO-331	RTCA Model-Based Development and Verification Supplement to DO-178C
FMEDA	failure modes, effects and diagnostic analysis
HAS	hardware accomplishment summary (DO-254)
HAZOP	hazard and operability analysis

HLR	high-level requirements (DO-178C)
IMA	integrated modular avionics
LLC	last-level cache
MCDC	modified condition / decision coverage (DO-178C Annex A)
MCP	multi-core processor (AC 20-193)
PSAC	plan for software aspects of certification (DO-178C)
SAS	software accomplishment summary (DO-178C)
SCM	software configuration management (DO-178C)
SDP	software development plan (DO-178C)
SMT	simultaneous multi-threading
SOI	stage of involvement (FAA software-certification audit)
SVP	software verification plan (DO-178C)
TQL	tool qualification level (DO-330)
UAS, UAM	unmanned aerial system / urban air mobility
WCET, WCRT	worst-case execution / response time

References

- [1] Dmitriev K, Zafar SA, Schmiechen K, et al (2020) A lean and highly-automated model-based software development process based on DO-178C/DO-331. arXiv preprint, arXiv:2010.06505, [2010.06505](https://arxiv.org/abs/2010.06505)
- [2] European Union Aviation Safety Agency (2022) Acceptable means of compliance AMC 20-193: Use of multi-core processors (annex i to ed decision 2022-001-r). EASA, Cologne
- [3] European Union Aviation Safety Agency (2024) Artificial intelligence concept paper issue 2: Guidance for level 1 & 2 machine learning applications. EASA, Cologne, concept-paper-stage guidance covering learning assurance, AI explainability, ethics-based assessment and human-AI teaming. Integration into acceptable means of compliance is the subject of Rulemaking Task RMT.0742.
- [4] Federal Aviation Administration (2004) Advisory circular AC 20-148: Reusable software components. U.S. Department of Transportation, FAA
- [5] Federal Aviation Administration (2022) Advisory circular AC 20-152A: Development assurance for airborne electronic hardware. U.S. Department of Transportation, FAA, provides COTS-component and previously-developed- hardware guidance, including service-experience considerations for COTS hardware and COTS IP.
- [6] Federal Aviation Administration (2024) Advisory circular AC 20-193: Use of multi-core processors. U.S. Department of Transportation, FAA
- [7] Federal Aviation Administration (2024) Roadmap for artificial intelligence safety assurance. U.S. Department of Transportation, FAA, roadmap establishing guiding principles for AI in airborne systems; precedes the issuance of any binding advisory circular on airborne AI/ML.

- [8] Gracioli G, Alhammad A, Mancuso R, et al (2015) A survey on cache management mechanisms for real-time embedded systems. *ACM Computing Surveys* 48(2):32:1–32:36. <https://doi.org/10.1145/2830555>
- [9] Green Hills Software (2024) INTEGRITY-178 tuMP RTOS enters verification phase for multicore DO-178C DAL A assurance on Intel Tiger Lake. Green Hills Software news release, URL https://www.ghs.com/news/20240617_Mercury_IntelTigerLake_Integrity-178_tuMP.html
- [10] Jin X, Chen H, Wang X, et al (2009) A simple cache partitioning approach in a virtualized environment. In: *Proc. IEEE Int. Symp. on Parallel and Distributed Processing with Applications (ISPA)*, Chengdu, Sichuan, China, pp 519–524, page-coloring in the Xen VMM, x86, transparent to the guest OS
- [11] Kim H, Rajkumar RR (2017) Predictable shared cache management for multi-core real-time virtualization. *ACM Trans on Embedded Computing Systems (TECS)* 17(1):22:1–22:27. <https://doi.org/10.1145/3092946>
- [12] King T (2014) Managing cache partitioning in multicore processors for certifiable, safety-critical avionics software applications. In: *Proc. 33rd IEEE/AIAA Digital Avionics Systems Conference (DASC)*, pp 8C3–1–8C3–7, <https://doi.org/10.1109/DASC.2014.6979540>
- [13] Kloda T, Solieri M, Mancuso R, et al (2019) Deterministic memory hierarchy and virtualization for modern multi-core embedded systems. In: *Proc. 25th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pp 1–14, <https://doi.org/10.1109/RTAS.2019.00009>
- [14] Li Y, West R, Missimer E (2014) A virtualized separation kernel for mixed-criticality systems. In: *Proc. 10th ACM SIGPLAN/SIGOPS Int. Conf. on Virtual Execution Environments (VEE)*, URL <https://www.cs.bu.edu/~richwest/papers/vee2014.pdf>
- [15] Mancuso R, Dudko R, Betti E, et al (2013) Real-time cache management framework for multi-core architectures. In: *Proc. 19th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pp 45–54, <https://doi.org/10.1109/RTAS.2013.6531078>
- [16] Martins J, Pinto S (2023) Shedding light on static partitioning hypervisors for ARM-based mixed-criticality systems. In: *Proc. 29th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, <https://doi.org/10.1109/RTAS58335.2023.00011>, 2303.11186
- [17] Ottaviano D, Ciraolo F, Mancuso R, et al (2024) The Omnivisor: A real-time static partitioning hypervisor extension for heterogeneous core virtualization over MPSoCs. In: *36th Euromicro Conference on Real-Time Systems (ECRTS 2024)*, *Leibniz International Proceedings in Informatics (LIPIcs)*, vol 298. Schloss

- [18] Pellizzoni R, Betti E, Bak S, et al (2011) A predictable execution model for COTS-based embedded systems. In: Proc. 17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), pp 269–279
- [19] Pradhan A, Ottaviano D, Jiang Y, et al (2025) Arm DynamIQ shared unit and real-time: An empirical evaluation. In: Proc. 31st IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), pp 241–254, <https://doi.org/10.48550/arXiv.2503.17038>, 2503.17038
- [20] Rapita Systems Ltd. (2024) A(M)C 20-193 vs. CAST-32A: What the change means for your DO-178C multicore project. Industry guide (white paper), URL <https://www.rapitasystems.com/downloads/amc-20-193-vs-cast-32a-what-change-means-your-do-178c-multicore-project>
- [21] RTCA / EUROCAE (2000) DO-254 / ED-80: Design assurance guidance for airborne electronic hardware. RTCA, Inc., Washington DC
- [22] RTCA / EUROCAE (2011) DO-178C / ED-12C: Software considerations in airborne systems and equipment certification. RTCA, Inc., Washington DC
- [23] RTCA / EUROCAE (2011) DO-330 / ED-215: Software tool qualification considerations. RTCA, Inc., Washington DC
- [24] SAE International (2010) ARP4754A: Guidelines for development of civil aircraft and systems. SAE International, Warrendale, PA
- [25] Saleab M, Sax F, Schumann J, et al (2021) Toward timing and memory analysis for UAS flight code. In: AIAA SciTech 2021 Forum, p 1918, <https://doi.org/10.2514/6.2021-1918>
- [26] Saleab M, Sax F, Schumann J, et al (2024) Low-level memory and timing analysis of flight code for unmanned aerial systems. Aerospace Systems 7(2):209–225. <https://doi.org/10.1007/s42401-024-00279-0>
- [27] Scolari A, Bartolini DB, Santambrogio MD (2016) A software cache partitioning system for hash-based caches. ACM Trans on Architecture and Code Optimization (TACO) 13(4):57:1–57:24. <https://doi.org/10.1145/3018113>
- [28] Stabellini S (2024) Xen project’s progress toward safety certification: Targeting ISO 26262 ASIL D and IEC 61508 SIL 3. Public record of the Xen Project Safety Working Group, reported through the ELISA (Enabling Linux in Safety Applications) community, July 2024, <https://elisa.tech/blog/2024/07/10/xen-projects-progress-toward-safety-certification/>; Embedded Open Source Summit keynote, April 2024, <https://events.linuxfoundation.org/embedded-open-source-summit/>;

- Xen Project wiki on safety certification, https://wiki.xenproject.org/wiki/Safety_Certification
- [29] Stevens BL, Lewis FL, Johnson EN (2015) Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems, 3rd edn. John Wiley & Sons
 - [30] SYSGO GmbH (2021) SYSGO introduces DO-178C / ED-12C DAL A certification kit for hypervisor-based real-time operating system PikeOS 5.1. SYSGO press release, URL <https://www.sysgo.com/press-releases/sysgo-introduces-do-178c-ed-12c-dal-a-certification-kit-for-hypervisor-based-real-time-operating-system-pikeos-51>
 - [31] Tabish R, Pellizzoni R, Mancuso R, et al (2023) X-Stream: Accelerating streaming segments on MPSoCs for real-time applications. Journal of Systems Architecture 138:102857. <https://doi.org/10.1016/j.sysarc.2023.102857>
 - [32] Wind River Systems (2024) Mitigation of interference in multicore processors for A(M)C 20-193. Wind River technical white paper, URL <https://www.windriver.com/sites/default/files/2024-10/656381-mitigation-of-interference-in-MC-processors.pdf>
 - [33] Xen Project (2024) Xen Project safety certification working group – scope, roadmap and artefact catalogue. Xen Project community wiki, https://wiki.xenproject.org/wiki/Safety_Certification
 - [34] Xen Project (2026) Requirements introduction document. <https://xenbits.xen.org/docs/latest/fusa/reqs/intro.html>
 - [35] Yarom Y, Ge Q, Liu F, et al (2015) Mapping the Intel last-level cache. IACR Cryptology ePrint Archive, Report 2015/905, URL <https://eprint.iacr.org/2015/905>, 2015/905
 - [36] Yun H, Yao G, Pellizzoni R, et al (2013) MemGuard: Memory bandwidth reservation system for efficient performance isolation in multi-core platforms. In: Proc. 19th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), pp 55–64, <https://doi.org/10.1109/RTAS.2013.6531079>
 - [37] Yun H, Mancuso R, Wu ZP, et al (2014) PALLOC: DRAM bank-aware memory allocator for performance isolation on multicore platforms. In: Proc. 20th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), pp 155–166, <https://doi.org/10.1109/RTAS.2014.6925999>
 - [38] Züpke A, Bastoni A, Chen W, et al (2024) MemPol: polling-based microsecond-scale per-core memory bandwidth regulation. Real-Time Systems 60:369–412. <https://doi.org/10.1007/s11241-024-09422-8>