DDC-I 2026 1986 40 Years
Safety Critical Software Solutions
For Mission Critical Systems

Safety-Critical Software Integration and Reuse in Avionics

Overview & Purpose

This paper argues that architectural modularity is the key enabler of cost-effective, safety-critical avionics development. Developing software to DO-178C DAL-A — the highest level of design assurance for aviation — typically costs tens of millions of dollars per component. By designing for a 'certify once, deploy anywhere' paradigm, development organizations can reduce recertification costs by an estimated 50–70% on derivative systems. The paper draws on over 20 years of operational experience with the DDC-I Deos™ RTOS to illustrate the broader architectural principles, which apply across avionics, automotive, defense, and other safety-critical domains.

The Layered Architecture

Safety-critical avionics software is organized as a layered stack, each layer exposing a well-defined interface to the layer above. This decomposition maps directly to DO-178C objective allocation and determines which components can be reused across programs:

- **Base Platform (RTOS & Drivers):** The certified Real-Time Operating System provides time and space partitioning per DO-178C and ARINC 653, ensuring critical and non-critical tasks coexist safely on the same hardware.
- **Middleware & Networking:** Deterministic network protocols — ARINC-664/AFDX, Time-Triggered Ethernet (TTE), and Time-Sensitive Networking (TSN) — decouple software components from network topology, a precondition for reuse.
- **Application Layer:** Custom application packages designed around standard APIs (e.g., ARINC 653 APEX) can be reused across platforms without rewriting, allowing test and verification artifacts to transfer with the component.
- **Legacy/Third-Party Software:** Pre-existing certified assets offer the highest reuse value. If built against the same RTOS and API set, porting may require only reintegration and regression testing.

RTOS Requirements: Determinism, Portability & Evidence Reuse

The RTOS is the most consequential design choice in a safety-critical architecture. It must be certified to the highest DAL required by any application in the system. Key capabilities include: (1) deterministic time partitioning via a two-level scheduler synchronized across all cores; (2) standardized ARINC 653 APEX and POSIX APIs enabling source-code portability; and (3) a dynamic linking model that allows certified binaries to be reused without recompilation. Critically, FAA Advisory Circular AC 20-148 provides a regulatory framework for Reusable Software Components (RSCs), allowing pre-certified binaries and their DO-178C evidence to be accepted by airworthiness authorities without full re-execution of verification activities — the regulatory foundation of the cost-reduction model.

Multicore Processors: Challenges & Mitigations

Multi-core processors (MCPs) offer attractive SWaP advantages but introduce interference between cores sharing cache, memory bus, and DRAM. This 'noisy neighbor' effect inflates worst-case execution time (WCET) estimates and creates certification risk. The regulatory landscape has evolved significantly: EASA's AMC 20-193 (January 2022) and the FAA's AC 20-193 (January 2024) superseded the earlier CAST-32A guidance and now permit dynamic task allocation across cores, provided the applicant demonstrates predictable, analyzable behavior. Mitigation strategies include software-configurable cache partitioning (isolating each partition to dedicated cache), distributed kernel object management per core to avoid global locking, and time-window scheduling aligned across all cores to prevent staggered cache access patterns. Sensitivity analysis toolsets that measure actual

WCET non-intrusively are essential for generating the interference analysis evidence required under A(M)C 20-193.

Economics of Binary Reuse

Verification accounts for 60–70% of DO-178C DAL-A development cost. Source-code portability reduces platform dependency but still requires compilation, integration, and regression testing — producing a new binary that must be re-verified. Binary reuse, enabled by dynamic linking and externalized XML configuration, allows certified components and their entire DO-178C evidence package to be reused with only integration-level testing for the new context. Integration timelines for derivative platforms can be compressed from 18–24 months to 6–9 months, saving hundreds of engineering-years of verification effort across a multi-variant aircraft family.

Emerging Challenges & Future Directions

The paper identifies four significant trends shaping the future of safety-critical software reuse:

- **AI/ML Certification:** The modular RTOS architecture enables AI/ML workloads to be deployed in lower-assurance partitions today, preserving safety-critical partition integrity until regulatory frameworks mature.
- **Supply Chain Resilience:** The dynamic-linking and hypervisor architecture localizes hardware changes to the RTOS or hypervisor layer and isolates library updates to individual modules, constraining the re-verification scope and mitigating national security supply chain risks.
- **Rust as a Safety Language:** The component-based architecture is language-agnostic: Rust, Ada, and C components share the same dynamic-linking and binary-reuse mechanisms once a qualified toolchain is available.
- **Cross-Industry Applicability:** The architectural principles — RTOS-based partitioning, binary reuse, open standards — apply directly to Urban Air Mobility, autonomous vehicles (ISO 26262), industrial systems (IEC 61508), and space flight software.

Key Conclusions

- A DO-178C DAL-A certified RTOS with ARINC 653 partitioning, dynamic linking, and externalized XML configuration is the non-negotiable foundation for all higher-level reuse.
- Binary reuse delivers the highest return on investment; source-code portability alone does not eliminate re-verification costs.
- FACE conformance and MOSA compliance align technical architecture with DoD acquisition requirements, enabling cross-program, cross-service component sharing.
- Multicore certification under A(M)C 20-193 requires cache partitioning, deterministic scheduling, and non-intrusive WCET analysis tooling.
- A certified Type-1 hypervisor enables mixed-criticality integration and hardware-generation independence, reducing SWaP and certification burden.
- Security and safety are complementary: component-level isolation supports both DO-178C evidence integrity and targeted binary obfuscation.

You can read the full paper at: https://www.ddci.com/content-request-safetycriticalreusepaper/

Presented at the Vertical Flight Society's 82nd Annual Forum & Technology Display, West Palm Beach, FL, May 5–7, 2026.