LYNX

# LYNX MOSA.ic Certification that Compounds

Whitepaper

# 1 Executive Summary

Most avionics platforms are optimized for first flight. But a program's success is judged far more on what it becomes than on how it starts: the first iteration of an aircraft is typically the least capable it will ever be, with its decisive capability arriving through successive upgrades. The money, the schedule risk and the program-level decisions therefore live somewhere else entirely — in the decades of re-approval that follow.

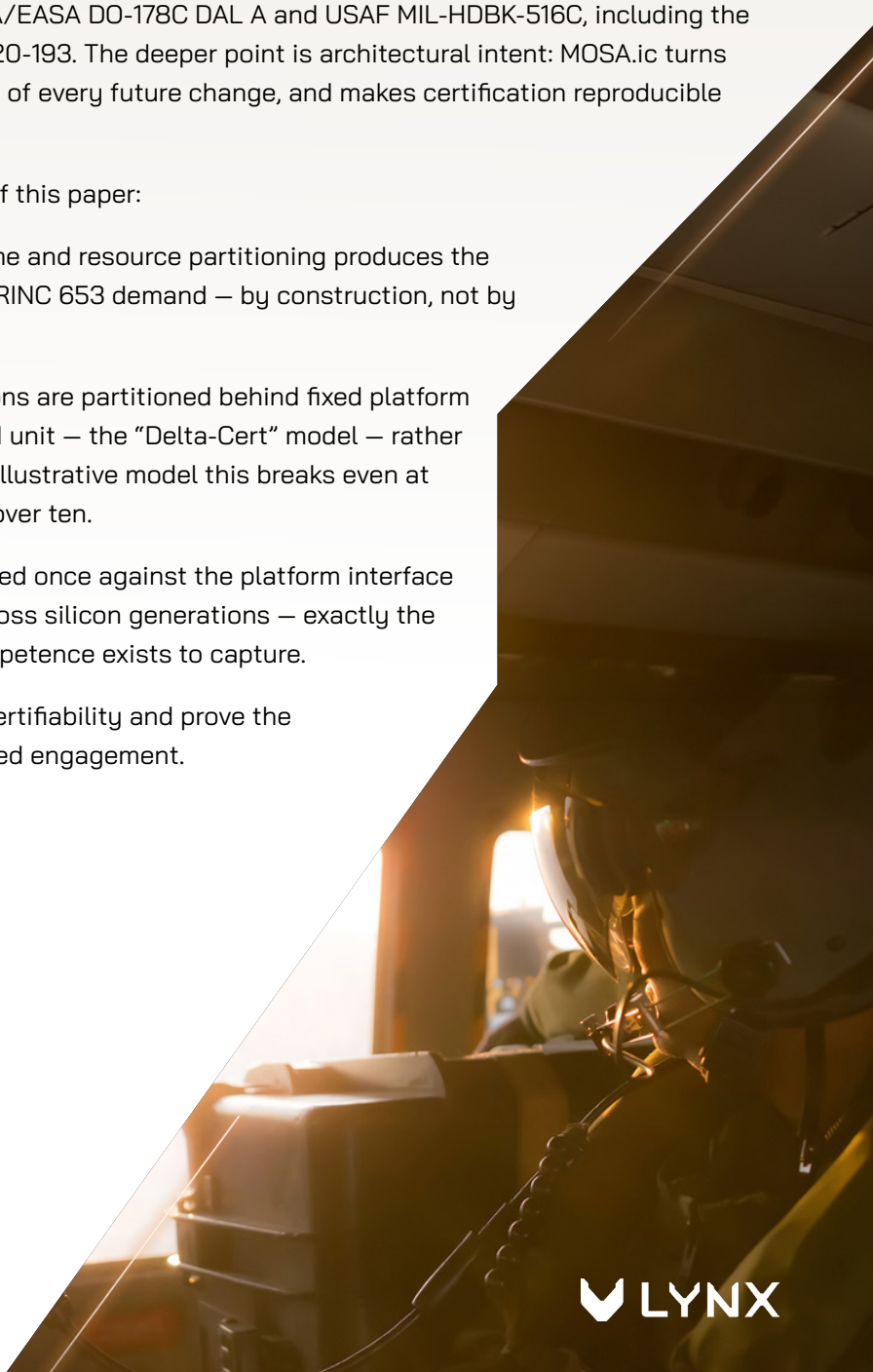
In safety- and security-critical aerospace, certification is not a downstream formality — it is the deliverable that unlocks every other outcome. The aircraft does not fly, revenue does not recognize, and capability does not field until a certification authority accepts the evidence. Across a 25-to-30-year airframe, the cost of producing and re-producing that evidence dwarfs the cost of writing the software in the first place. Obsolescence alone guarantees it: over that lifespan nearly every electronic system will be replaced at least once as parts reach end-of-life, and each replacement re-opens the certification argument even when the function is unchanged. Certification is not a one-time event — it recurs for the life of the platform, and its cost compounds with every change.

LYNX MOSA.ic is built for this challenge. It is a configurable partitioning system — a type-1 separation kernel with a configurable software backplane — developed to FAA/EASA DO-178C DAL A and USAF MIL-HDBK-516C, including the multicore safety guidance now formalized as A(M)C 20-193. The deeper point is architectural intent: MOSA.ic turns isolation into certification evidence, bounds the cost of every future change, and makes certification reproducible across programs rather than re-earned each time.

Three consequences follow, and they are the spine of this paper:

- **Isolation Becomes Evidence** - Robust space, time and resource partitioning produces the non-interference arguments that DO-178C and ARINC 653 demand — by construction, not by after-the-fact analysis.
- **Change Becomes Predictable** - Because functions are partitioned behind fixed platform interfaces; a change re-verifies only the changed unit — the "Delta-Cert" model — rather than re-opening full-system certification. On an illustrative model this breaks even at the first change and runs roughly 58% cheaper over ten.
- **Certification Becomes Portable** - Evidence earned once against the platform interface is reusable across airframes and re-hostable across silicon generations — exactly the leverage a cross-program System Centre of Competence exists to capture.

MOSA.ic is an architecture whose figure of merit is certifiability and prove the economics on your own numbers through the bounded engagement.



## 2 Certification that Compounds

An architecture's real figure of merit in this domain is not raw capability. It is how cheaply, repeatably and durably it produces approvable evidence. Three truths define why certification compounds, and every architectural choice should be measured against them.

### Certification is the Gate

No approval, no deployment. Schedule slips, capability delays and revenue recognition all hang on the certification authority's acceptance. In practice the certification basis — not the codebase — is the critical path of a safety-critical program.

### Certification is the Cost

Initial development is the down payment. Over the service life of an airframe, sustainment, obsolescence refresh and repeated recertification dominate total cost of ownership. Any architecture that treats certification as a one-time event is optimizing the wrong variable.

### Certification Recurs

Every change re-opens the safety argument. That has always been true; what is new is that the industry's forced migration to multicore raised the difficulty of the argument precisely where everyone must now go. The guidance path from CAST-32A to the formalized A(M)C 20-193 exists because shared-resource interference on multicore silicon is genuinely hard to bound. Whoever can produce a repeatable multicore interference argument owns a decisive advantage; whoever cannot inherit open-ended schedule risk.

## 3 Why Traditional Architectures Compound Cost, Not Evidence

The classical operating environment fuses business logic to a proprietary runtime and a proprietary application binary interface, on top of a kernel that also owns resource allocation, scheduling and drivers. It also fails to compound — for five structural reasons:

- **Security** - For mixed criticality systems the probability of losing control of the processor to a DAL D/E environment is high. Architectures that do not strictly isolate application data services from physical resource management capabilities are very risky.
- **Compounding Coupling** - In multicore environments, resource-arbitration and data-forwarding services often compete for execution time and state. Hosting them together in a traditional kernel makes timing and data integrity increasingly difficult to manage, raising the risk of cascading failures. Modern architectures should isolate platform services in independent execution environments.
- **Proprietary Binding** - Applications are bound to a specific vendor's ABI and runtime. Components are not interoperable, and their certification evidence is not portable — it is captive to one stack.
- **Platform Extensibility** - There needs to be provisions in the platform architecture that allows integrators to respond to demand for platform features (e.g. data protection) over time while preserving critical properties of previously certified platform features.
- **Non-reusable Evidence** - Because nothing is cleanly bounded, verification artifacts cannot be reused. Every update pays for a broad regression and recertification. As the source brief puts it bluntly: NOT REUSABLE.
- **The Disaggregated Model** — Interoperable components over an open connection on a simple, common foundation — is the one that turns certification into a compounding asset rather than a recurring cost.



LYNX

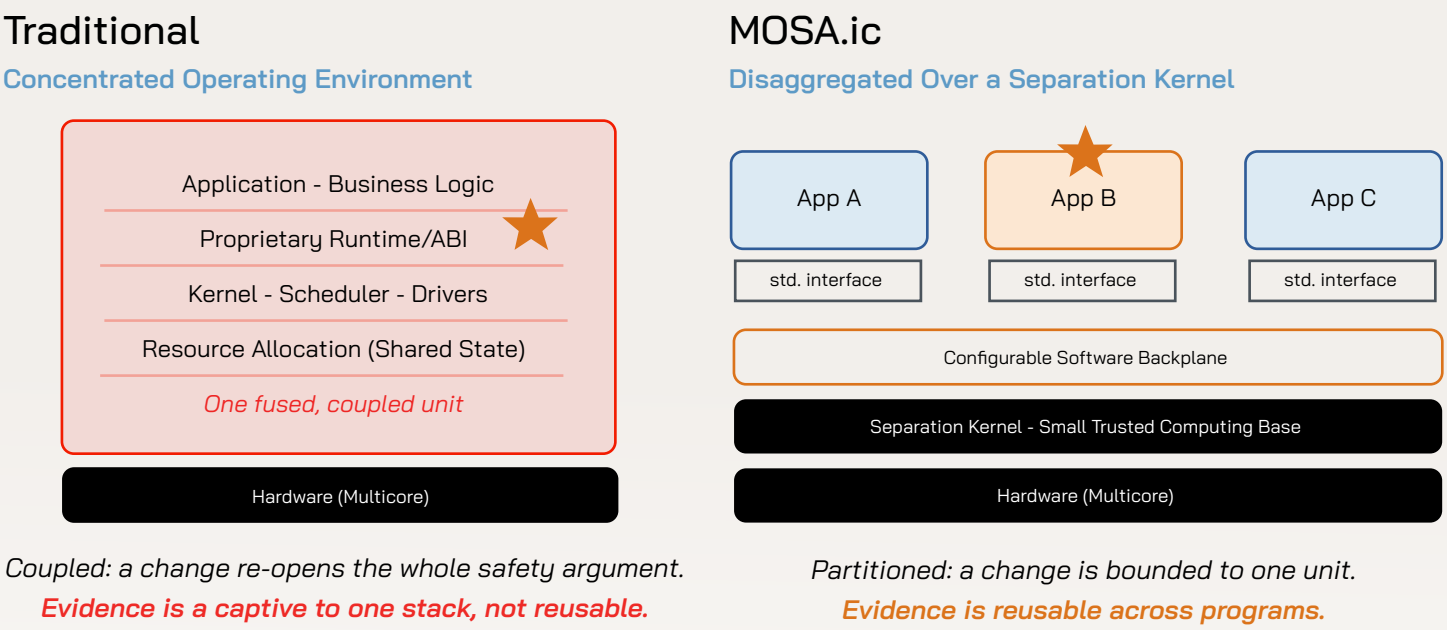


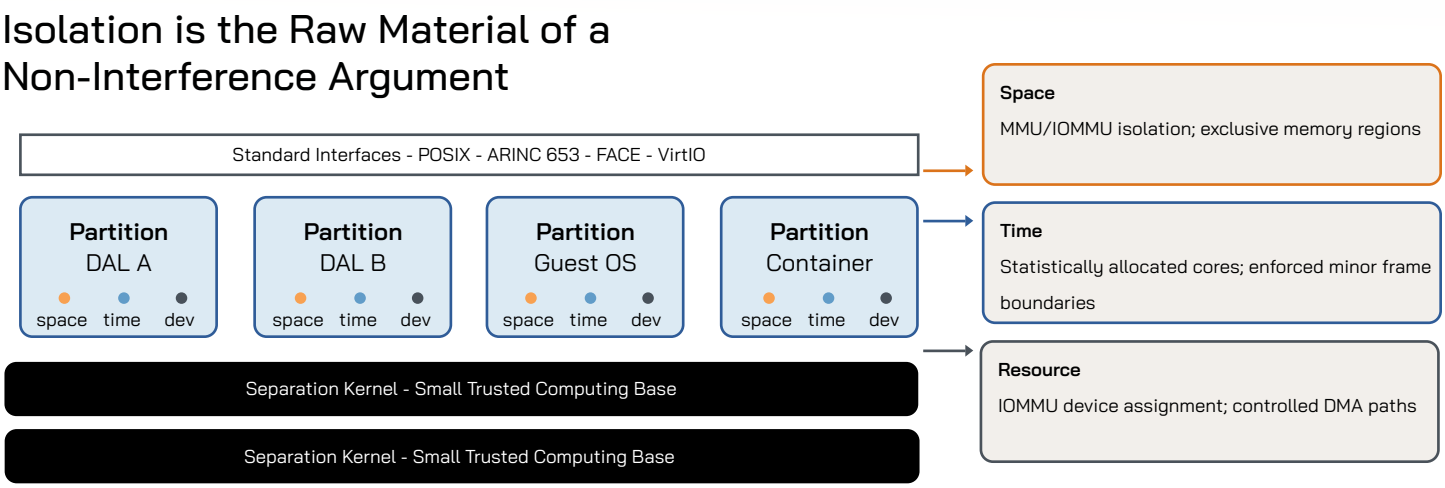

***Figure 1. Concentrated vs Disaggregated - Why Coupling Loses the Challenge***

## 4 Architected for Certification: How MOSA.ic Wins

Every MOSA.ic mechanism below exists in the source brief as a capability. The reframe is to read each one as an instrument for producing approvable evidence.

### 4.1 Isolation as Evidence

Certification for partitioned systems turns on one question: can you demonstrate that partitions do not interfere with one another in space, time or resources? MOSA.ic answers it by construction. Memory is partitioned with MMU/IOMMU isolation and three-level addressing (host-physical, guest-physical, guest-virtual), with exclusive regions by default and shared regions only where explicitly configured. Cores are statically allocated — virtual CPUs are pinned to physical cores and cannot migrate — and time is divided by a hierarchical scheduler with enforced minor-frame boundaries. Devices are assigned through the IOMMU. Each of these is not merely a feature; it is the raw material of a non-interference argument.




*Each boundary- space, time, and resource, is verified by construction, then closed by test.*

***Figure 2. Isolation as Evidence - Space, Time, and Partitioning Over a Small, Trusted Computing Base.***

### 4.2 A Small, Message-Based Trusted Computing Base

MOSA.ic is a true type-1 separation kernel with no shared kernel state and a message-based architecture between virtual machines. The certification consequence is direct: a smaller, simpler trusted computing base is a smaller thing to verify to DAL A, and a cleaner basis for a Multiple Independent Levels of Security argument. Platforms that grew partitioning onto a monolithic RTOS carry a larger critical core into every safety and security case they make.

### 4.3 Standard Interfaces Make Components Independently Certifiable

Applications build to standard interfaces — POSIX, ARINC 653, FACE, VirtIO, Vulkan/OpenGL SC — and are independent of the host kernel. That independence is what lets a component be certified as a unit and integrated without dragging its neighbors back into scope. It is also what makes the evidence portable, which Section 6 turns into a cross-program argument.

### 4.4 Multicore Interference: Mitigated by Design, Closed by Test

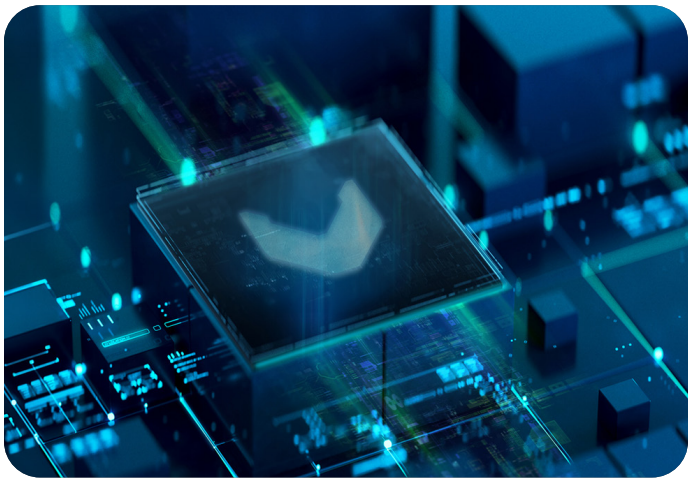
This is the hardest part of the modern challenge, and it is where the source brief is deepest. MOSA.ic pairs design-time mitigations with a matching verification battery so that each hazard has both a control and a proof:

| Design Mitigation | Verification/Analysis |
|---|---|
| Static core allocation; controlled operating configuration | Minor-frame boundary enforcement; cross-partition independence tests |
| DMA path control; memory and I/O protection | IPC functional verification; interrupt-latency measurement |
| Controlled interrupt routing; deterministic schedule | Partition-overrun enforcement + worst-case execution-time analysis |
| Fault containment | Fault-injection tests |
| Configuration integrity (SRP / PDI) | Configuration validation via the CheckSRP tool |

*The certification value is that the argument is systematic and repeatable, not bespoke to one integration — the property A(M)C 20-193 rewards.*

### 4.5 Configuration as a Controlled, Verifiable Artifact

The behavior of a MOSA.ic system is defined by its configuration — the System Runtime Package and its partitioning data, expressed as a binary configuration vector — rather than emerging from runtime interaction. Because the configuration is an explicit artifact, it can be checked (CheckSRP), qualified and version controlled. Certification prefers a system you can point at over a system you must observe.



## 5 Delta-Cert: The Economics of Bounded Change

The economic benefit is more specific than "partitioning isolates change." It comes from managing change at the configuration level: where a change is expressed as a bounded edit to the platform's configuration vector, and the expected result of that edit is known in advance, certification reduces to verifying that the system still operates within its established tolerances — rather than re-running the full-system argument. That is the substance of a "Delta-Cert": confirming a change is compatible with the platform's interfaces, resource allocations and characterized operating range, complementing DO-178C and ARINC 653.

This benefit is real only to the degree the platform earns it, and it carries a build cost. It depends on configurable platform services — networking, GPU multiplexing and the like — and, critically, on having characterized how configuration changes affect performance properties and on having established the configuration-range limits within which behavior is guaranteed. Lynx is building this configurable-services layer; it is not yet complete, and the performance-impact characterization and range limits are still being established. Where that characterization exists, change collapses to a bounded tolerance check; where it does not yet, a change still requires the analysis to bound its effects. The Delta-Cert economics therefore scale with platform maturity — which is itself an argument for engaging now and shaping that characterization around your target configurations.

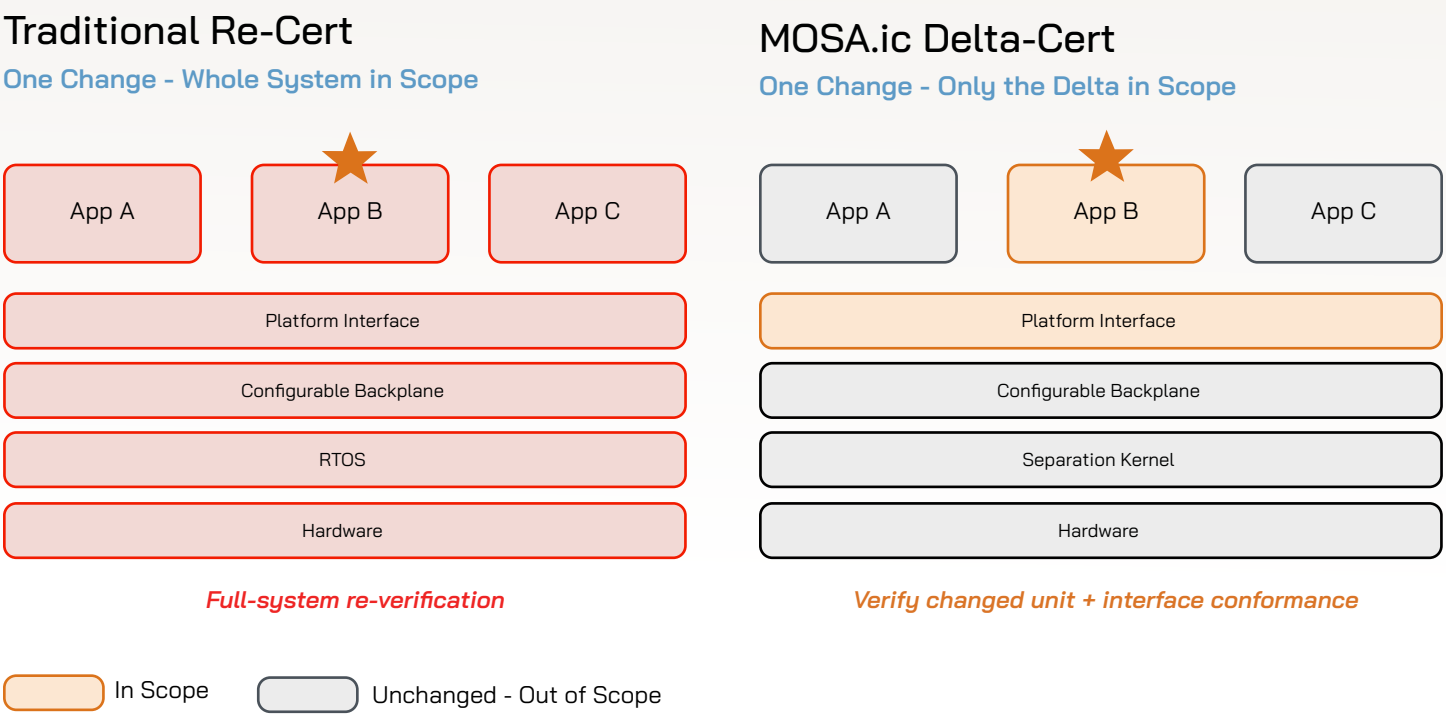
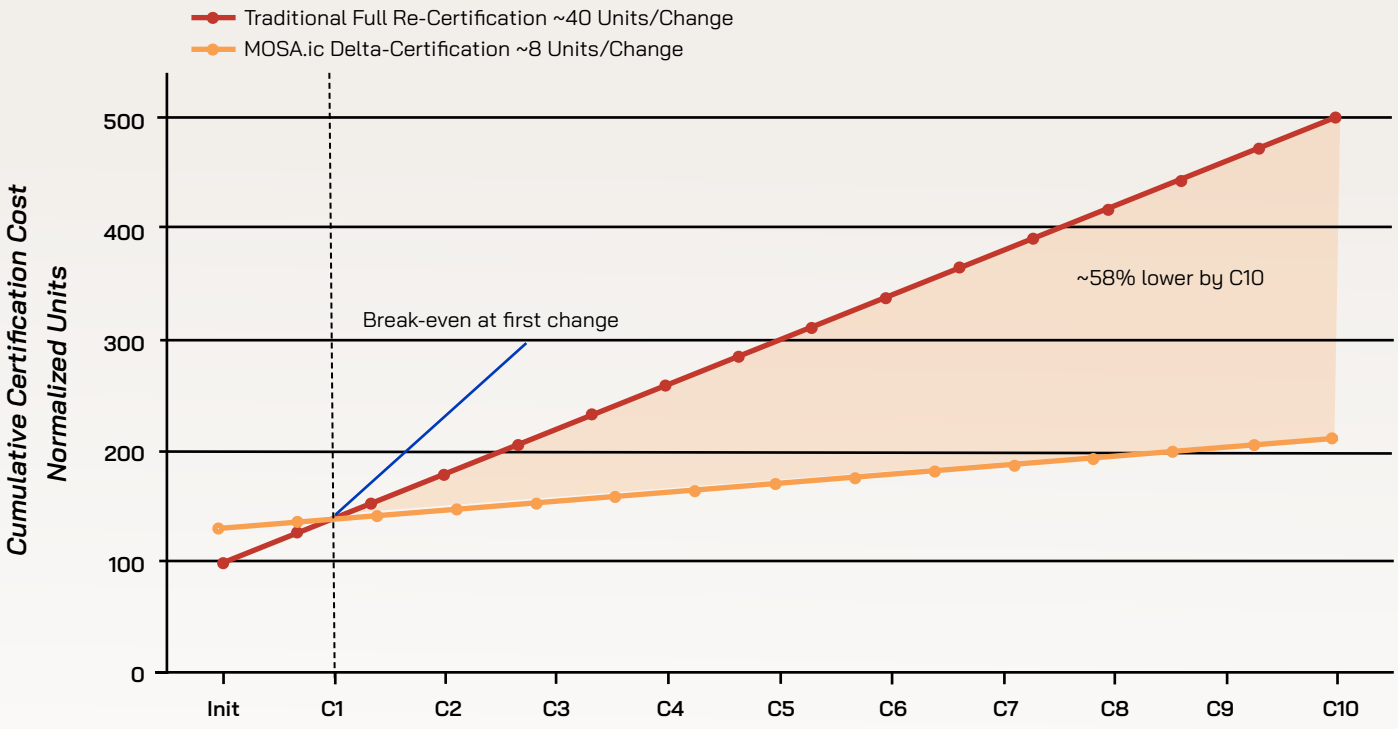



*Figure 3. The Scope of Recertification — Full-system Re-verification Versus a Bounded Delta.*

The economic shape of that difference is the whole case. The chart below model's cumulative certification cost across a sequence of change events for a traditional full-recert approach versus Delta-Cert. Figures are illustrative and normalized (see Appendix A); the shape, not the absolute numbers, is the argument.




*Illustrative Model - Cumulative Certification Cost, Normalized Units*

Three things to read off it:

- An Upfront Premium Recovered Immediately - Establishing the certified partitioning platform costs a little more at the start (~30% in the model). That premium is recovered by the first change event.
- A Structurally Lower Cost of Change - Each subsequent change is roughly an order of magnitude cheaper because its scope is bounded (~8 vs ~40 units).
- A Gap that Widens for the Life of the Platform - At ten changes the cumulative saving is about 58%. Real programs see far more than ten changes over three decades, so the divergence only grows.

This is the mechanism by which a certification-driven architecture converts a technical property — partitioning — into program economics .

These are Lynx model figures, not measured metrics: normalized units chosen so the shape — not the absolute numbers — makes the point. The mechanism is architectural; the magnitudes redraw around your program's actual recertification effort.

# 6 Reproducing Certification Across Programs

The most valuable certification evidence is the evidence you never have to produce twice. Delta-Cert bounds the cost of change within one program. The same partitioning discipline extends the benefit across the portfolio and across time.

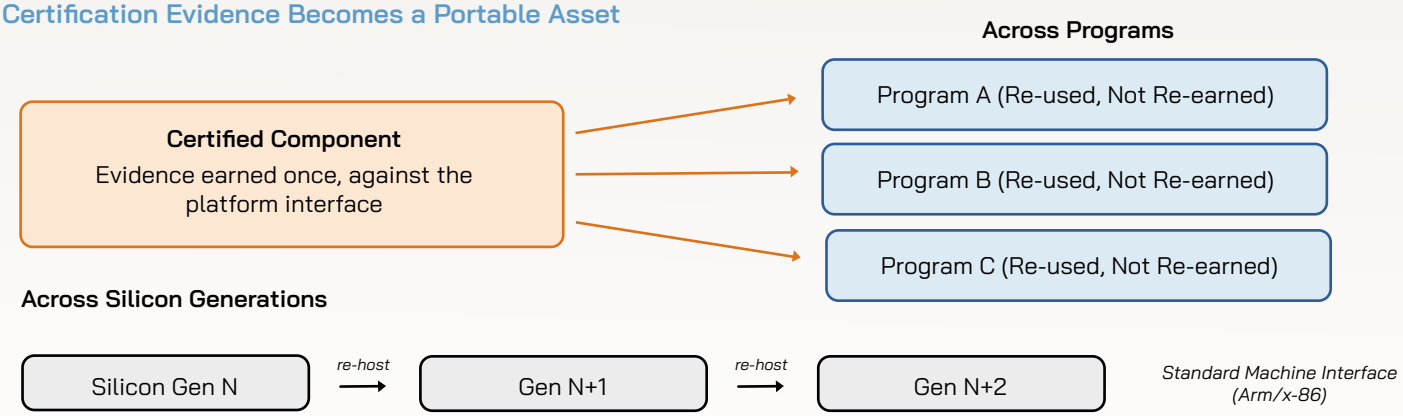
### Across Programs — Reuse and Amortization

Partitioning makes reuse creditable, not automatic: the platform fixes each item's interfaces and boundaries so its evidence can transfer, but the credit realized depends on the item and on staying within the usage domain it was accepted against. The organizational task is to engineer items for reuse — well-bounded and specified to a domain wide enough to span the fleet — so credit follows by design."

### Across Silicon — Obsolescence Without Re-Argument

The standard machine interface (Arm v8-A / x86) decouples the application lifecycle from the silicon lifecycle. Over a 30-year platform the underlying part will reach end-of-life more than once; MOSA.ic's supported hardware family spans Intel, AMD/Xilinx and NXP/Arm devices at varying maturity. When a part ages out, certified applications re-host onto new silicon without a rewrite — and the interface stability keeps the re-hosting inside a bounded, delta-style argument rather than a fresh full-system campaign.

### Certification Evidence Becomes a Portable Asset




Certify once; reuse across the portfolio and re-host across obsolescence cycles - without a fresh full-system campaign.

***Figure 4. Certify Once, Re-Use Everywhere - Portable Evidence Across Programs and Silicon Generations.***

# 7 The Assurance Package

What MOSA.ic brings to a certification campaign, beyond the architecture itself.

| Domain | Basis Addressed |
|---|---|
| Airworthiness safety | FAA/EASA DO-178C to DAL A; A(M)C 20-193 multicore guidance (CAST-32A successor) |
| Military airworthiness | USAF MIL-HDBK-516C and associated airworthiness criteria |
| Security | NIST Risk Management Framework; MILS-style domain isolation |
| Application interface | ARINC 653 Part 1 & 2; POSIX.1-2008 real-time & pthread; FACE OSS Safety Base / Extended |
| Graphics | OpenGL SC1/SC2; Vulkan SC |

*Standard and certification statuses evolve; confirm current specifics against Lynx's certification artifacts before citing externally.*

### A Structured Verification Strategy

The verification approach is staged so that evidence accumulates in a defensible order: (1) inputs before execution — reference hardware configuration, representative SRP/PDI configuration, timing expectations and instrumentation; (2) nominal baseline tests under no stress; (3) high-load and stress-condition tests; and (4) faulted-condition tests. Runtime coupling, network-stack coupling and queuing hazards each receive the same design-plus-verification treatment shown in Section 4.4.

### Reference Systems As Proof

The same certifiable primitive — separation kernel plus configurable backplane — is demonstrated across mission profiles in the source brief: IMA and the digital backbone, MILS, FACE, mixed-criticality consolidation, and cross-domain solutions, extending even to hardened containerized edge deployments. Breadth from a single certifiable core means a certification investment is leveraged, not siloed.

## 8 Why Now - The Window Is Forced

- **Multicore is No Longer Optional** - Single-core certified silicon is aging out, and with A(M)C 20-193 now the accepted basis, multicore certification is simultaneously required and finally tractable.
- **Obsolescence Forces a Platform Choice Now** - Parts certified 10–15 years ago are reaching end-of-life; silicon and OS decisions are unavoidable in this window — better made against a 30-year architecture than by re-federating.
- **Open-Architecture Mandates are Maturing** - MOSA and FACE-style approaches have moved from aspiration to procurement requirement. Set the certified reference while the standard is being defined, not after a proprietary stack is entrenched.
- **Connected and AI-Enabled Aircraft Need Designed-in Coexistence** - Hosting non-deterministic workloads beside flight-critical functions must be architected — and certified — from the start; it cannot be retrofitted onto a federated design.

## 9 Why Lynx Over The Alternatives

The competitive question is: whose architecture produces cheaper, more portable certification evidence?

| Differentiator | Certification Consequence |
|---|---|
| Partitioning-native, small TCB | A type-1 separation kernel with no shared kernel state is a smaller thing to verify to DAL A and a cleaner multicore-interference argument than partitioning bolted onto a monolithic RTOS. |
| Heterogeneous guest model | Unikernels, full guest OSes, ARINC 653 partitions and bare processes coexist — bring existing, already-certified images rather than rewrite to one vendor's model, lowering migration and re-cert cost. |
| Openness as leverage | Standard, non-proprietary interfaces let evidence and applications be multi-sourced and re-used, rather than captured by one ABI. |
| Ahead on edge / cloud convergence | Containers running beside DAL A functions position Lynx further along the connected-aircraft trajectory than classic IMA RTOS vendors. |

## 10 Start Now - The Platform is Ready

MOSA.ic is cert-ready across the two instruction-set architectures that matter for avionics compute — Arm and Intel / x86 — spanning 32- and 64-bit word sizes and both single-core and multicore topologies. Whether the target is a legacy single-core refresh, a 64-bit consolidation onto a modern multicore SoC, or something in between, the platform meets it with an established certification basis rather than a development effort.

| Architecture | Word Size | Core Topology | Availability |
|---|---|---|---|
| Arm | 32- & 64-bit | Single core & multicore | Cert-ready today |
| Intel / x86 | 32- & 64-bit (IA-32 / x86-64) | Single core & multicore | Cert-ready today |

*"Cert-ready" means the platform carries an established certification basis and supporting artifacts for these targets, ready to enter a program campaign. Exact device support, release maturity and available artifacts vary by target and assurance level; we confirm current coverage for your chosen silicon as the first step of any engagement.*

### What That Means For You

- **You Start Against a Proven Baseline** - Effort goes into your applications and your program's certification basis — not into inventing the substrate beneath them.
- **The Long-Pole Risk is Already Retired** - The multicore interference argument of Section 4.4 — the hardest, longest-lead item of a modern campaign - arrives with design mitigations and a verification approach in hand.
- **Your First Change is a Delta-Cert, Not a First Cert** - You inherit a platform whose isolation, small, trusted computing base and configuration control are already structured to produce approvable evidence.

## Appendix: Delta-Cert Cost Model — Assumptions and Method

These figures are illustrative and normalized. They are chosen to show the shape of the economics — the break-even point and the widening gap — not to assert absolute costs. Replace them with program actuals before any external or decision-making use.

### Model

- **Unit** - normalized cost (one unit ≈ the cost of certifying one bounded change under Delta-Cert).
- **Traditional** - initial certification = 100 units; each change event = 40 units (full-system regression/recert scope). Cumulative after N changes = 100 + 40N.
- **Break-even** - 100 + 40N = 130 + 8N → 32N = 30 → N ≈ 0.9 — Ex: the upfront premium is recovered by the first change event.
- **At N = 10** - Traditional = 500, Delta-Cert = 210 → ≈58% cumulative reduction, widening with every further change

### Parameters to Replace with Program Data

To make this defensible for a specific program, substitute: the expected number of change events over the service life; the average certification effort of a full-system recert versus a bounded delta in that program's DAL mix; the applicable labor and authority-liaison rates; and any tool-qualification amortization. The model structure (fixed initial cost plus per-change cost, compared across two per-change slopes) remains the same.

**When Can We Begin?**

The short answer is “now”.

Tell us your target architecture and certification basis, and we will map it to the supported family, stand up a reference platform against it, and walk our certification evidence against your program — starting whenever you are.

edge@lynx.com
www.lynx.com/solutions/lynxmosaic-ai