jamf

Why **Purpose-Built** Beats Good Enough — **Every Time.**

Choosing a generic Unified Endpoint Management (UEM) for Apple is like staffing your on-call rotation with people who've never touched the stack. They'll figure it out, but your SLA won't survive the learning curve.

Here is how Purpose-Built Apple Management beats a generic UEM for every outcome:

Outcome	Generic UEM	Purpose-Built Apple Management
New Apple feature support	New Apple platform capabilities are deprioritized against a cross-platform backlog; **your team waits quarters**, not days, to manage what Apple just shipped.	**Management controls** for new Apple capabilities, like Apple Intelligence, Managed Device Attestation and Declarative Device Management, are built and tested before your org needs them.
Employee onboarding	**Onboarding requires IT intervention** to close enrollment gaps, push missing configs or troubleshoot stalled setups. New hires wait. Tickets get opened. First impressions take a hit.	**New hires unbox their device**, sign in and are fully set up with apps, settings and access before lunch on day one. IT never touches the device.
Admin workflow efficiency	**Cross-platform abstraction** layers sit between the admin and the OS; what works today may silently break after an Apple update, and there's no vendor who owns that gap.	**Apple-native frameworks** mean admins configure, enforce and report on exactly what Apple exposes: no fidelity lost in translation, no undocumented behavior to chase down.
Security posture	**Security policies are retrofitted across platforms** to find the lowest common denominator; Apple-specific controls are underutilized and audit exceptions accumulate.	**Compliance frameworks map directly to Apple's built-in security controls.** CIS benchmarks, platform SSO, hardware attestation and Managed Device Attestation work as Apple designed them.
Support quality	**Support is generalized across platforms**; Apple-specific issues get triaged by people who manage many ecosystems, and escalation paths are longer when the issue lives deep in Apple frameworks.	**Every support engineer works with Apple exclusively:** they've seen your problem before, they know the platform. Resolution comes faster because no one is context-switching from a Windows ticket.
Cost predictability	**Costs spike predictably every fall.** Re-testing, re-certifying and rebuilding broken workflows after Apple updates is an annual tax your IT team pays in overtime and deferred projects	**Support costs stay flat** across Apple release cycles. The vendor absorbs the complexity so your team doesn't have to.
Apple feature completeness	**Features are scoped** to what works across every platform the vendor supports; Apple gets a subset, and the delta between what Apple built and what you can actually manage grows over time.	When Apple ships a capability, you get the whole thing: **full configurability and the ability to enforce policy** the way Apple intended.

Every Apple OS release is a test

A purpose-built, best-of-breed solution passes it on day one.
A generic UEM makes your IT team pay for it in overtime.

The real cost of good enough for your Enterprise

- Good enough **to deploy** → **until the workarounds break.**
- Good enough **to enroll** → **until the tickets pile up.**
- Good enough **to check the compliance box** → **until your best admin leaves because they're tired of fighting the tool.**
- Good enough **to put in the board deck as cost savings** → **until the cost shows up everywhere else.**

jamf

www.jamf.com

Ready to make the transition from good enough to Best-of-Breed Apple Management? Trust Jamf, the leader in Apple management and security.