ATARC powered by GovExec

WHITE PAPER

Modernizing RMF for Continuous, Evidence-Based Security

Treat Authorization as an Engineering Function, not a Compliance Milestone

Prepared by Dr. Darren Death, EXIM, and the ATARC cATO Working Group

The fastest way to move RMF away from compliance and into the mission space is to stop treating authorization as a milestone and start treating it as a continuous engineering process. RMF shouldn't be a side activity, it should be embedded in how a system is built, deployed, and maintained. That starts by tying every security control to a clearly defined, mission-relevant risk—and making sure senior leadership owns the decision to mitigate, transfer, or accept that risk. Care must be taken to ensure that risk ownership does not drift downward; while responsibilities can be delegated, authority cannot. Too often, lower-level individuals assume they can accept risk or bypass controls without executive awareness. That is a governance failure. Just as important, security controls must be written into engineering and development requirements as mandatory—not treated as optional guidance. When security controls are presented as negotiable, they are often deprioritized or ignored entirely under pressure to deliver. Security cannot be left to interpretation; it must be embedded as a condition of acceptance at every stage of system design and implementation.

From there, agencies need to build delivery pipelines capable of proving, on demand, that those risks are being managed. Once leadership understands that mission success depends on a current and accurate risk picture, documentation stops being produced separately. It becomes an automatic output from the telemetry already protecting the system. The traditional System Security Plan becomes a relic—it's no longer a document you write, it's a dataset your Governance Risk and Compliance system delivers via a dashboard. Evidence for a system is structured, machine-readable, and built into the same pipelines that validate security and operational readiness. What remains as documentation are operational plans—contingency, incident response, recovery—but those aren't compliance artifacts. They're real management tools for execution. Their value isn't in how they're formatted; it's in how effectively they help teams act under pressure. If a document isn't used to reduce downtime, coordinate response, or support mission continuity, it's overhead, not security.

This shift doesn't require waiting on future policy. The current environment supports it. OMB's FY25 FISMA guidance encourages agencies to replace static artifacts with operational metrics powered by automation. More importantly, it states clearly that

RMF shouldn't be a side activity, it should be embedded in how a system is built, deployed, and maintained.

meeting metrics is not the objective—reducing real-world risk is. NIST's Cybersecurity Framework 2.0 reinforces this direction by elevating "Govern" to a core function and linking evidence directly to organizational risk decisions, not audit checklist completion. The NIST Secure Software Development Framework (SSDF) further strengthens this shift by defining secure development practices that are measurable, repeatable, and integrated into the delivery process—not bolted on afterward. It prioritizes automation, traceability, and engineering discipline over documentation volume. The Continuous Diagnostics and Mitigation (CDM) program reinforces this trajectory by requiring agencies to feed automated asset, vulnerability, and configuration telemetry into federal dashboards—converting system security postures into enterprise risk signals. FedRAMP, while still operating under its current framework, is also signaling a transition. The conceptual FedRAMP 20x initiative proposes accepting automated, machine-validated evidence and commercial validation models in place of traditional narrative documentation. While not yet implemented, this concept points the direction the government is moving in—toward automated assessment models, faster evidence cycles, and reduced audit overhead. Agencies that align with that direction today position themselves to benefit immediately when these and other capabilities become operational.

This shift also requires embedding enforceable security requirements directly into contracts, particularly for custom software development, COTS acquisitions, and system integration. Risk management cannot begin at deployment; it starts with procurement. Agencies must require vendors to align with secure software development practices defined in NIST SP 800-218 (SSDF) and codify those expectations in acquisition language. That includes secure coding standards, automated unit and integration testing, static and dynamic analysis, policy-as-code enforcement, and reproducible build processes. Software deliverables must include machine-readable SBOMs, vulnerability disclosure procedures, configuration benchmarks, and integration points for telemetry. These aren't compliance artifacts; they are data inputs for the pipelines that determine real-time authorization status. This approach is already supported by EO 14028 and OMB M-22-18, which direct agencies to obtain attestations for secure software practices and establish supply chain transparency through SBOMs and vulnerability disclosure policies. For contractors handling Controlled Unclassified Information (CUI), agencies must also enforce NIST SP 800-171 compliance and, where applicable, CMMC certification. These are not checkboxes—they are minimum conditions for participating in federal delivery pipelines. If security requirements aren't contractually binding at the outset, agencies will be forced to remediate after the fact—at greater cost, reduced visibility, and higher risk.

ATARC

Codify Controls and Eliminate Duplicate WorkData?

The starting point is the control baseline. Every NIST SP 800-53 control that your system implements should be tied to a definitive, automated test. That might be a unit test for infrastructure-as-code, a policy-as-code rule for cloud resources, a static or dynamic security check in the CI/CD pipeline, or runtime detection feeding a SIEM. For controls already enforced at the enterprise or cloud provider boundary, stop rewriting them—inherit and eliminate redundant documentation.

Where possible, output the results of these tests in OSCAL or another structured, machine-readable format. This evidence should be generated automatically by the system—not authored manually. Whether it's OSCAL, JSON, YAML, or a native export from your tools, the goal is the same: assessors should be able to consume the evidence directly, without requiring engineers to restate it for compliance. The industry is steadily moving toward OSCAL as the common format to reduce the overhead of transforming and managing bespoke evidence structures. Standardizing on OSCAL reduces integration friction, simplifies ingestion across tools, and enables evidence to flow from engineering pipelines into governance platforms without manual intervention.

If your GRC platform doesn't yet support OSCAL, you don't abandon the model—you adapt how evidence is managed. The objective remains: generate authoritative, structured security evidence at the system level. If your pipeline produces OSCAL but your GRC can't ingest it directly, intermediate formats like JSON, CSV, or YAML can serve as translation layers. The GRC system becomes a consumer of evidence—not the source. You're not asking engineers to reformat outputs manually; you're transforming existing telemetry into a form that the GRC can display and analyze. Long-term reliance on ad hoc transformations increases friction and maintenance cost. Standardizing on OSCAL, where feasible, strengthens toolchain interoperability and ensures control evidence can move cleanly across systems. The principle remains unchanged: the source of truth lives in the pipeline. GRC platforms should interpret that evidence—not recreate it.

The principle remains unchanged: the source of truth lives in the pipeline. GRC platforms should interpret that evidence—not recreate it.

Integrate RMF Into Agile Delivery

Next, stop treating RMF steps like separate documents or compliance gates. Collapse them into the sprint cadence already driving delivery. Instead of one-time data categorization exercises, use a dynamic data inventory that developers and engineers can query as part of normal system design. Categorization exists to define the potential impact of a breach—low, moderate, or high—so that the right set of controls can be applied up front based on what the system is protecting and adjusted over time as that protection need changes. Control selection should be driven by prepopulated baselines aligned to system impact levels and applied according to data protection requirements that evolve with the threat landscape—not revisited only during annual meetings. Implementation and assessment are part of the CI/CD pipeline. If a control test fails, the build fails. A POA&M entry opens automatically, complete with evidence. Authorization isn't a signature—it's a state tied directly to the health of the delivery process. When the pipeline passes, the system stays authorized. That turns ATO maintenance into a side effect of normal development, not a separate workflow. This approach is fully supported by NIST SP 800-160, which reinforces that security must be engineered into the system development life cycle from the start. It's not about proving compliance once—it's about maintaining assurance through continuous validation and operational feedback.

Cut Cost, Accelerate Delivery, Improve Outcomes

This approach reduces costs because security engineering and compliance are no longer separate efforts. Almost every dollar spent building secure systems also satisfies audit and authorization requirements. You're not funding parallel workflows, you're funding a single integrated pipeline that produces usable evidence as part of delivery. Documentation time drops because control evidence is delivered automatically, not manually rewritten to satisfy templates. Speed increases because risk decisions are based on current telemetry—not delayed by reviews of outdated artifacts. Security improves because controls are enforced continuously through automation, fail fast when broken, and generate actionable signals to alert of remediation. When risk management is built into how systems are procured, developed, and operated, outcomes improve—because security is no longer reactive or bolted on. It's part of how the system works.

Leadership Must Enforce the Model

None of this works without leadership enforcement. Agency executives must make it clear that pipelines are the authoritative source of risk data—and that security controls enforced by those pipelines are non-negotiable. These are not optional safeguards; they are tied directly to mandatory, executive-owned cybersecurity requirements. No team should be permitted to remove or bypass a control to save time without leadership visibility and formal risk acceptance. Once that accountability is in place, the rest is engineering: codify the controls, automate the evidence, output it in a structured format, and let live telemetry drive authorization decisions instead of compliance checklists.

Apply the Same Model to COTS, Infrastructure, and Cloud

The same approach applies across asset classes. Whether it's custom applications, COTS products, on-premise infrastructure, or cloud services, the principle is the same: treat each as a continuously instrumented supply chain component and plug it into the evidence pipeline.

For COTS, this begins at acquisition. Require vendors to deliver machine-readable SBOMs, maintain a vulnerability disclosure cadence, and provide hardened configuration benchmarks mapped directly to your security control baseline. These requirements should align with recognized industry standards like DISA STIGs, CIS Benchmarks, or vendor-secured baselines, and must be enforceable through automated configuration management and vulnerability scanning. As soon as a COTS product is installed—whether on an endpoint, server, or virtual image—tooling should verify cryptographic signatures, scan for known CVEs, and validate system configurations against enterprise policy. This can be done through existing asset management, endpoint protection, or continuous monitoring platforms. Because the evidence is structured, it can be embedded in an OSCAL component definition and inherited by any system using the product.

For on-premise infrastructure, control enforcement begins with how the environment is provisioned and configured. Every server, switch, hypervisor, or appliance should be deployed using infrastructure-as-code or automated configuration tooling that reflects your approved security baselines. A single "script" should be capable of building the secure state—and serving as both the implementation and the control evidence. System hardening should follow pre-approved templates aligned with NIST 800-53 and supported by industry benchmarks such as DISA STIGs or CIS. Once deployed, the infrastructure should be continuously assessed through configuration management and vulnerability scanning tools that validate patch status, software inventory, service exposure, and unauthorized changes. That evidence, when structured, can be expressed in OSCAL and inherited by any authorization boundary that includes the asset.

For cloud services, control implementation begins with identifying what's inherited from the provider and what must be enforced by the agency. Most baseline controls—encryption at rest, infrastructure patching, physical safeguards—are implemented by the provider and validated through FedRAMP authorization. But the process of consuming that evidence is still manual. While OSCAL is emerging as the preferred standard for delivering control implementation, most providers today deliver their packages in PDFs or proprietary formats. Agencies must extract what's usable, align inherited controls to their baseline, and focus implementation on what remains. Until OSCAL or another machine-readable format becomes the norm, agencies will need to normalize and map provider documentation manually.

Cloud also presents unique challenges for continuous review. Limited visibility, proprietary architectures, and inconsistent telemetry make it difficult to fully validate provider-implemented controls in real time.

Concepts like FedRAMP 20x are moving the government toward a model where machine-readable, accessible security data serves as authoritative proof of control effectiveness. Once implemented, this model will allow agencies to inherit the majority of their baseline automatically and focus validation efforts on the controls they still own.

ATARC

Telemetry First, Not Text First

Security must be built, measured, and enforced as part of system delivery—not documented after the fact. When control tests are codified, automated, and tied to live telemetry, they produce the same risk data executives already rely on to run the mission. That alignment drives down cost, accelerates delivery, and replaces static artifacts with real-time assurance. But it only works when leadership treats security controls as mandatory—not optional. If control tests can be removed without executive visibility, risk ownership has broken down. When control results are tied directly to governance outcomes, risk becomes visible, actionable, and accountable. The frameworks, policies, and tooling to support this model already exist. What's needed now is execution. It's time to stop managing documentation and start managing the system.