WHITE PAPER

FedRAMP 20X and the Automation of Arrested Development

Nate Demuth

CALFIRE

"The King is dead! Long live the King!"

This is how I feel as I listen to the announcements for the vision of FedRAMP 20X and goals to automate everything everywhere. But why? In the world of AI, stars, and horseshoes, clovers and blue moons, shouldn't I hear such things and fall in prostration? You would think so. Unfortunately, a career as an architect, engineer, advisor, integrator, and operator in the field of cloud and cybersecurity, several years spent with Cloud Service Providers (CSPs) navigating FedRAMP, as well as a childhood of chasing leprechauns to gold at the end of the rainbows, has taught me better.

FedRAMP 20X aims to create efficiencies in the assessment process, and they're further breaking it down into four (4) components and working groups (Continuous Monitoring, Automating Assessments, Applying Existing Frameworks, Continuous Reporting). These groupings, detailed **here** and below, are not mutually exclusive. In this article, we'll touch on all aspects to some degree, but we will primarily focus on continuous monitoring, assessment, and reporting. The stated goals of FedRAMP 20X as it relates to these domains are:

- **Continuous Monitoring:** "In the new FedRAMP, the PMO will no longer force industry cloud service providers to upload raw vulnerability scan logs to a central FedRAMP service, and then process log files to generate reports manually. Cloud service providers will instead generate reports directly using their own automation systems against a simple standard, and then make those reports available to customers via normal channels."

- **Automated Assessment:** "FedRAMP 20x is focused on building a cloud-native, automated security assessment process that enables continuous innovation. This means that instead of evidence consisting of screenshots, we want evidence provided continuously based on actual configurations."

- **Continuous Reporting:** "This [is a] future where ongoing risk monitoring is enforced, validated and reported continuously. In this future, CSPs will have a transparent, effective methodology for reporting overall risk posture and incident status via standard customer channels without FedRAMP in the middle. Continuous reporting may involve the following areas:

    - **Data generation:** data is automatically generated from system components (CI/CD, monitoring agents, vulnerability scans, configurations, etc.)

    - **Data analysis:** data is normalized and aggregated to transform into information in the form of Key Security Indicators (KSIs) that represent system risk

    - **Reporting:** Information is presented to customers in consolidated reports that reflect near-real time risk posture, enabling informed, risk based decision making"

True reform is needed regarding these aspects of FedRAMP, and many of us are here and ready to provide proposed solutions (which we'll be covering at length in this article). However, if we're going to make meaningful and lasting impacts, we must take it seriously (even if radical), which these proposed changes don't seem to do. This means stepping back and asking:

1. What are the outcomes we're trying to achieve, and how do we measure success (i.e., goals and objectives)?
2. What led us to doing things the way we do today (a Chesterton's Fence analysis – understanding the past before acting)?
3. What would the proposed solutions and/or changes look like, and for who?
4. Why are we saying our proposed changes will be any better than the current way (e.g., a cost benefit analysis between options, including the "do nothing")?

So, buckle up, grab a cup of milk and your favorite sugar infused breakfast grains, and let's walk through these topics.

"Questions, So Many Questions"

Before we rush into what a potential "automated everything" architecture and operation could look like, let's start by exploring some of the above questions. There is a lot of common ground shared here between federal agencies, the FedRAMP program managers, CSPs, Elon Musk, Third Party Assessing Organizations (3PAOs) and advisors, and even Lucky the Leprechaun. So, we'll start there.

What are the goals and objectives of the current Continuous Monitoring program? To ensure that CSP's are maintaining an adequate level of risk protections for their offerings to federal agencies, they are required to provide monthly reports (including outputs from ongoing technical scans) that demonstrate compliance to control requirements. They must also undergo annual audits which conduct deeper dives into the broader compliance posture. Through each activity (monthly and annually), they must report any gaps, including plans to remediate or justification of a waiver. Success is measured by compliance with controls.

This is what the proposed changes are targeted at reforming. The goal is valid, the objectives for how to perform it are tough to swallow and have been the bane of existence for many for years.

What is the stated problem by the recent FedRAMP 20X announcement? The approach and methods for Continuous Monitoring (including monthly scanning and reporting) and annual assessments are archaic; driven by time and cost consuming manual processes that are systematically suppressing innovation and technological advancement of government agencies to achieve their mission (which was the whole point of FedRAMP in the first place).

This is about as poetic a statement you'll ever hear as an architect and engineer who's lived the FedRAMP journey.

What is FedRAMP PMO hoping to get out of proposed changes? To maintain or enhance existing quality of risk protection at a reduced time and cost by eliminating manual processes and associated work products that don't contribute to verifiable confidentiality, integrity, and availability (CIA) of government systems.

Things just got Shakespearean.

What is it FedRAMP PMO is going to do to achieve this with FedRAMP 20X changes? They're telling "industry" to figure out a way to implement mechanisms to automate everything, everywhere, for ongoing monitoring and assessment of compliance with technical control standards and report it to a (or many?) dashboards.

Annnnnnnd, it's gone.

There's a litany of questions related to the who, what, when, where, and how as it relates to execution (which we'll ask and try to steel-man answers for in the next section). Those questions aren't the problem for me though.

The real problem is: why did we decide that automating the existing approach and methods was the right solution in the first place?

1. Is it possible the process itself (manual or automated) is the problem?
2. Are configuration checks on technical standards and reviewing of Common Vulnerability and Exposures (CVEs) really the best and most efficient means by which to validate CIA (regardless of what the standard is)?
3. Is reporting compliance really the best measure of success?
4. Is it possible that automating the existing approach and methodology will just reallocate existing costs of time and effort into other forms?
5. How do we rationalize this objective with the other broader FedRAMP 20X objectives?
6. Rather than automate a broken system, what if we flipped the model entirely, eliminating pointless documentation and validating real-world resilience instead?

In other words, it's not what was proposed that's bothersome. It's what they seemingly didn't stop to ask themselves (or others) before they proposed it.

"Fools rush in where angels fear to tread." In this case, fools automate what shouldn't exist, then act surprised when it fails faster.

By not asking these questions, there are already a series of striking contradictions purely at a strategic level. For example, part of FedRAMP 20X is to remove the rigidity of control standards to allow CSPs more autonomy to innovate how they implement the requirements. And yet, automation requires rigidity and standardization to work in the first place (let alone scale). How are we going to allow CSPs to "do whatever works," but then expect them to achieve standards against an automated continuous monitoring, assessment, and reporting?

Will the required rigidity for the new goals end up working backwards to place CSPs and government into the same or comparable constraints (and therefore costs) as the ones we have now related to control rigidity? Forget Shakespeare, this has the makings of a Greek tragedy on the level of Oedipus.

The next section we'll explore these foundational issues as I attempt to genuinely construct approaches for how to meet the FedRAMP 20X vision.

"Steel-manning a Potential Architecture and Operations"

To get to a potential architecture and operations that meets the FedRAMP 20X objectives, we need to anchor the requirements against another set of questions first:

- Who is responsible for what in this new model?
- What constitutes "automated," and what exactly are we assessing and reporting?
- How often would these "things" take place?
- Where will we accept scans from, and where will those go?
- How are we going to facilitate all this?

There's surely more to be asked, but these cover enough ground for discussion. These also loosely mirror the agendas of the current running working groups.

We also need criteria to evaluate options given the various ways in which this could take shape (not provided as part of the current working groups). As we established before, **the purpose here is to maintain or enhance risk protection quality while reducing time and cost by eliminating processes that don't support the CIA of government systems (remember, that is the ACTUAL goal).**

Let's be more specific though. The proposed solution should consider:

1. Cost to build, integrate, manage and scale the solution. Within this are costs for people (automation architects, engineers, and operations teams), process (key activities and programs to facilitate), and technology (new code development, software licensing, hosting, storage, data transfer, etc.) all require thought.
2. The time to build, integrate, test, manage and scale the solution.
3. The ability to ensure the same level of technical FedRAMP policy (or comparable) is assessed and validated.

As we evaluate these variables, it's important to also try to conceive of unintended, emergent consequences which will fall outside the standard balance sheet. Not to design perfection, but to consider whether there are intuitive outcomes that would directly undermine what we're trying to optimize for in the first place.

For example, of particular interest is how the solution will impact organizations of different scale. Will the benefits of the proposed solution be biased toward different types of CSPs (e.g., CSPs with one ATO vs many, CSPs with simple vs complex environments, CSPs with one vs many agencies, etc.)? Given a stated objective is to make things easier for small-to-mid market players, this is important.

These types of emergent consequences have been historically ignored and represent some of the more harmful patterns that exacerbate the core inefficiencies.

Now that that is settled, let's begin with potential shared responsibility models.

"Who is responsible for what in this new model?"

There will need to be a new shared responsibility model constructed to facilitate the proposed automated continuous monitoring, assessment, and reporting. Because this could permeate dozens of ways, we'll create a few scenarios that represent the relative extremes for the purpose of discussion.

Option	Own & Managing Entity	Validation Entity	Integration Burden	Economies of Scale	Key Risks
Option 1	CSP	3PAO	High on CSP	Low (per CSP)	Fragmented tools; meta-validation burden; high barrier for small CSPs
Option 2	3PAO	3PAO	Shared (CSP + 3PAO)	Moderate (per 3PAO)	Integration complexity; platform fees; innovation constraints
Option 3	Government	3PAO	High on CSP	High (centralized)	High government cost; potential bureaucratic sprawl; reduced agility

Each model reshuffles the burden, but none eliminate it.

- **Option 1** burdens every CSP; results in my new favorite daytime drama, "Desperate Dashboard."
- **Option 2** consolidates tooling at the 3PAO level but still creates integration and economic friction.
- **Option 3** centralizes it all – offering theoretical scale – but at the likely cost of agility and fiscal sanity.

Realistically, the future lies somewhere between Option 1 and 2 – balancing CSP ownership with third-party standardization. But no matter the model, the shift to continuous monitoring will demand a rethink of roles, incentives, and where we draw the lines of trust.

"What constitutes “automated,” what are we scanning for, and when?"

The current guidance that has been provided is that the automation will be specific to technical policy. It calls for us all to define a means by which to perform configuration checks against that policy. There is also the requirement (implied at least) for the results to funnel into a dashboard for ongoing viewing as opposed to having manual monthly POA&M reports submitted and reviewed.

Let’s start with technical policy controls first. Again, given our objective is to maintain or enhance risk posture, we’re assuming there won’t be a reduction in these standards.

Here’s a list of the primary technical control families and example controls within each – these are the ones typically considered "technical":

Access Control (AC)

- AC-2: Account Management
- AC-3: Access Enforcement
- AC-4: Information Flow Enforcement
- AC-5: Separation of Duties
- AC-6: Least Privilege
- AC-7: Unsuccessful Login Attempts
- AC-11: Session Lock
- AC-17: Remote Access
- AC-18: Wireless Access
- AC-19: Access Control for Mobile Devices
- AC-20: Use of External Information Systems
- AC-21: Information Sharing
- AC-22: Publicly Accessible Content

Audit and Accountability (AU)

- AU-2: Audit Events
- AU-3: Content of Audit Records
- AU-6: Audit Review, Analysis, and Reporting
- AU-8: Time Stamps
- AU-11: Audit Record Retention
- AU-12: Audit Generation

Identification and Authentication (IA)

- IA-2: Identification and Authentication (Organizational Users)
- IA-4: Identifier Management
- IA-5: Authenticator Management
- IA-8: Identification and Authentication (Non-Organizational Users)

System and Communications Protection (SC)

- SC-7: Boundary Protection
- SC-8: Transmission Confidentiality and Integrity
- SC-12: Cryptographic Key Establishment and Management
- SC-13: Cryptographic Protection
- SC-17: Public Key Infrastructure Certificates
- SC-18: Mobile Code
- SC-19: Voice Over Internet Protocol
- SC-28: Protection of Information at Rest
- SC-29: Protection of Information in Transit
- SC-32: Information System Partitioning
- SC-34: Non-Mandatory: Fail in Known State
- SC-39: Process Isolation
- SC-42: Sensor Capability and Protection

System and Information Integrity (SI)

- SI-2: Flaw Remediation
- SI-3: Malicious Code Protection
- SI-4: System Monitoring
- SI-7: Software, Firmware, and Information Integrity
- SI-10: Information Input Validation
- SI-11: Error Handling

These are the core technical controls. There are also configuration management, media protection, and maintenance controls with some technical components, but the list above captures the main "technical" body.

Some of these controls have a relatively clear path for a simple configuration check. However, many do not, as they are fundamentally achieved via general design decisions (such as segmentation, Public Key Infrastructure design, system monitoring, etc.). There are possibly a few ways to tackle this challenge:

- **Automated Design Document Review –** CSP's must provide a standard design document with design choices related to non-configuration related aspects of technical policy. The document (rather than the environment) is then scanned to determine compliance with the expected design principles. This OSCAL 2.0, could be integrated with the dashboard along with configuration policy checks.

- **Agent Smith + Architecture Mapping –** Another option could be for AI agents to be allowed access into an environment to crawl through and vet the environment's design layout and following some set of criteria for identifying patterns of concern which get flagged. It's possible this could also be augmented with some form of environment mapping solution(s) off the marketplace. The agent could maybe even do the configuration checks itself while it's at it or just augment what is produced by configuration scanning tools. We'll ignore for now the fact that auto-generated memes of cats are cratering OpenAI's SLAs, and pretend that nuclear energy / GPU's / and other highly scarce resources of the world are prioritized for FedRAMP.

Regardless of which configuration we go with, things like asset and state management, and compute and storage costs associated with scan processing will be where most of our design and management efforts are going to concentrated.
Additionally, the decisions around shared responsibility model (Option 1 vs 2 vs 3) will have significant implications for this as well. If CSPs own and manage their systems (Option 1), timing is relatively less impactful (though a CSPs scale may change this). If Option 2 or 3 are taken though, then we pay for the centralized or partially centralized model with having to account for wild system performance demands and the associated architecture and operational costs to support it.

"Where to Implement Assessment, and Where to Send Findings"

Source Environments: Production, Staging, and CI/CD Integration

To be honest, this isn't just a question of "where do we point the scanner and collector?" – it's a question of where reality lives, and what counts as evidence.

In the envisioned world of FedRAMP 20X, automated continuous monitoring, assessment, and reporting aims to replace ad hoc reports and annual rituals with real-time or near-real-time evidence pipelines. This requires a fundamental rethinking of where assessments are sourced from, how often they occur (primarily addressed in the previous section), and where they are reported.

To accommodate the diversity of cloud service architectures and operational maturity across CSPs, a tiered scanning model may be most pragmatic:

- **Production environments: This would represent the gold standard –** capturing the true, deployed state of systems exposed to real-world conditions. However, operational sensitivity and risk-aversion may limit full production scanning, especially for intrusive assessments.

- **Staging / pre-prod mirrors:** Many CSPs already maintain pre-production environments that "closely mirror" production configurations. Scanning here offers a balance between realism and safety and can be a required proxy in cases where production scanning is constrained.

- **CI/CD pipelines and IaC templates:** A forward-looking vision would also incorporate "left-shifted" scans – evaluating infrastructure-as-code (IaC), container manifests, and configuration scripts before they're deployed. This not only enables preventive controls but also supports automation within the DevSecOps lifecycle.

To achieve preservation of risk control validation, we'd probably want a hybrid model, with primary reliance on staging environments validated for parity (at least that's how we imagine the direction to go), and optional incentives for production-level validation where feasible.

"Where Do Results Go?"

A Technical Trust Center (or TTC), as I'm calling it; built and managed according to which ever shared responsibility model we are running with.

To unify and streamline visibility, the outputs of automated assessments would be aggregated into a standardized, trusted reporting destination. This TTC would:

- **Ingest** configuration and compliance evidence from CSP-managed or third-party scanners.

- **Normalize** that evidence into machine-readable formats.

- **Present** dashboards to agencies, FedRAMP PMO, and (potentially) 3PAOs for oversight.

- **Maintain** integrity through digital signatures, time-stamped submissions, and immutable logs.

The other part of this question is also what exactly do we show in terms of metrics (also vaguely called out in FedRAMP 20X)? There are a few ways we could approach this:

- **Purely Compliance:** Just control lists with status indictors.

- **Meta Compliance:** e.g. Total vulnerabilities, total asset coverage, newly introduced vulnerabilities, # and % of flags resolved over time, # and % of findings my asset type (code, server, DB, etc.).

- **Risk Posture:** Calculation of potential risk criticality based on theoretical probabilities and control postures.

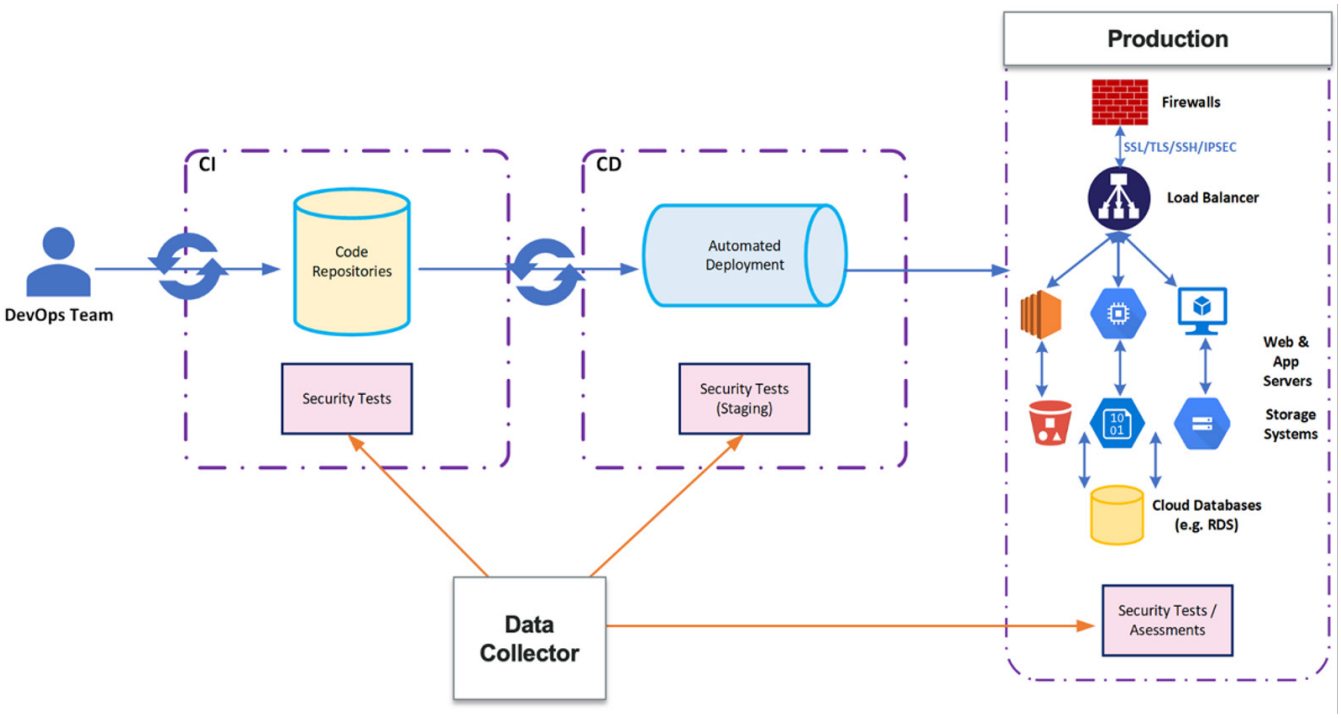
We would need to decide on what exactly the agencies want to see (and hopefully they would all agree on the same specifications). Today, they effectively get all three via the current process. If we design the TTC to mirror this, we may effectively be just reinventing a good old Governance Risk and Compliance (GRC) platform. This again has different implications based on which shared responsibility model we run with; not just for CSPs, 3PAOs, and agencies, but for my stock portfolio in GRC software.

"DevSecOps: Not Just a Buzzword"

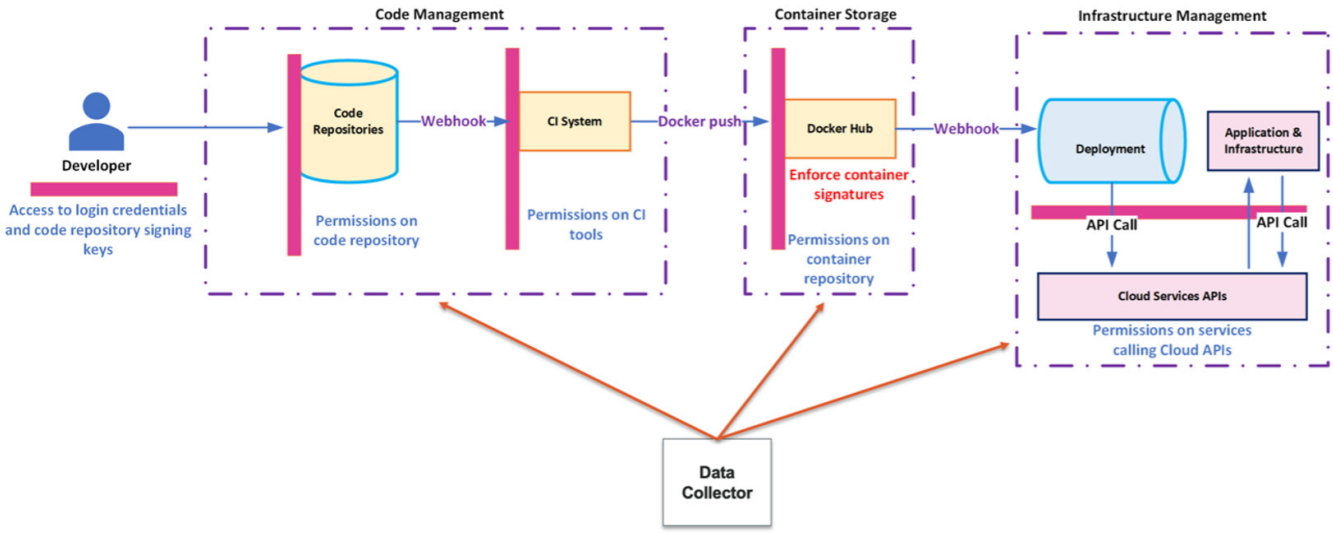
Automated continuous monitoring, assessment, and reporting does have some nice alignments naturally with DevSecOps principles (so it makes sense to see the PMO recommend it be an area for consideration). In the best-case scenario:

- **Security scans are embedded into CI/CD workflows,** with pass/fail gates tied to policy compliance.
- **Findings are exported automatically** into the Technical Trust Center without human intervention.
- **Version-controlled compliance baselines** ensure drift is visible over time, even across microservices or ephemeral deployments.

Below is a notional model for what the collectors could pull from to achieve this at both the development and meta development (i.e., CI/CD hardening) levels.



- **Test Driven Security (TDS):** Testing against security policies integrated into CI/CD or dev pipelines to run security tests again security policies / requirements ahead of deployment into production infrastructure
- **Unit Testable Security Requirements:** Controls should be defined down to small, specific, unit testable modules; vague requirements such as "encrypt network communications must be avoided. Good example is "enforce HTTPS with ciphers X, Y, Z on all external traffic ports."
- **Fail and Correct Early:** Missing of security controls / requirements are detected early and fast, allowing developers to fix issues as part of daily development activities

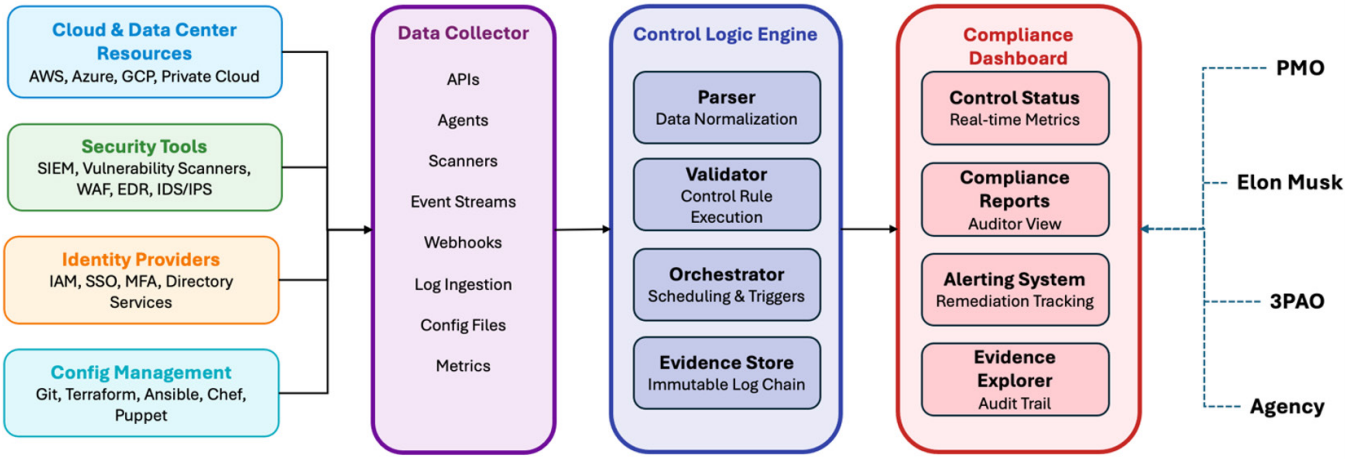


- **CI/CD Access and Least Privileges** for interactions between CI/CD components protected at each layer via access and least privileges.
- **Code Commit Tags** leverage code commit signatures and tags, and write scripts to review / audit those signatures outside CI/CD pipeline.
- **Docker Container Trust:** Signatures to evaluate how Docker Trust Signatures could help bring increased trust and validation.
- **Secrets and Key Manager:** Distribute secrets to systems securely using specialized tools like HashiCorp Vault, never store in cleartext at rest.

There's a lot of big assumption here regarding CSPs being in a place of CI/CD maturity where the prerequisites of these integrations are readily in place (let alone that all CSP's use the same container architectures and standards). There is also a question of how this works within the constraints of the certification boundary and requirements that today often break CI/CD maturity within FedRAMP. This level of increased compliance would drive significant costs for small and large CSP's alike. Add it to the shopping cart, I guess.

"All Together Now"

The following represents a notional architecture for much of what we've covered in this section.



There's several pros and cons to evaluate across these options. It also serves as a nice mapping of software and infrastructure firms I'll be making investments into if this initiative moves forward. Regardless of the specific approach taken, each configuration ends with the following types of issues (for those not keeping score):

1. **Continued False Sense of Security**
    - Automation may detect surface-level compliance without capturing deeper architectural flaws or dynamic threats.
    - High "pass" rates on config checks can obscure real security gaps if attack simulation is absent.
2. **Production Blind Spots**
    - If production scanning is optional or avoided, the data driving risk decisions may be based on environments that lack real-world fidelity.
3. **CI/CD Integration Isn't Universal**
    - Many CSP environments do not use modern CI/CD or Infrastructure-as-Code still to this day.
    - Mandating DevSecOps-style integration risks penalizing providers who haven't fully modernized or have meaningful barriers preventing them to do so.

4. **Complexity of Trust Center Management**

    - A centralized Technical Trust Center (TTC) introduces substantial governance, uptime, and security concerns as another attack surface.

    - If managed by government, it requires high investment and carries operational risk. If third-party managed, it introduces trust dependencies and vendor lock-in potential.

5. **Data Overload and Alert Fatigue**

    - Continuous scans across multiple environments and pipelines can generate massive volumes of data, making it difficult for agencies to triage what matters.

6. **Standardization vs. Flexibility Paradox**

    - Automation depends on standardization. Yet FedRAMP 20X also promises CSP flexibility and innovation. These are in tension – automating compliance may unintentionally reintroduce rigid requirements via backdoor.

7. **Cost Shifting, Not Cost Saving**

    - While manual reporting may decrease, CSPs may bear increased costs in integration, telemetry generation, evidence formatting, and coordination with 3PAOs or government.

8. **Security of the Security System**

    - The TTC, scanner agents / collectors, and CI/CD evidence pipelines become high-value targets for attackers. The system that validates trust must itself be secure – and auditable.

9. **Not All Controls Are Automatable**

    - Many NIST controls (e.g., segmentation, policy enforcement, encryption implementation) resist simple yes/no automation.

    - Attempting to force automation into these areas can lead to either superficial checks or massive implementation complexity.

10. **Large Burden on Small CSPs**

    - The costs of participation, integration, and technical complexity disproportionately affect small and medium-sized providers – potentially consolidating the FedRAMP ecosystem around larger players who will spend a lot as well to implement, but have the cash.

The result: we've automated ourselves into a comparable, if not greater, state of mission failure related to total cost and time to achieve and maintain ATOs, while producing benefits that are biased against the same parties the changes are intended to support.

It's hard to walk away from even the steel-manning of this approach without feeling like I'm stuck in the Wizard of Oz; solutions that lack brains, heart, and courage.

But what if started from an entirely different premise? One where we start with the fundamental so "why" we even test what we test today.

We'll explore something that follows this path in the next section.

"Less Controls, More Crash Test Dummies"

I once read a fortune cookie that said, "Great Leaders are Great Simplifiers."

If the federal government is willing to consider the wisdom of ancient deserts, there's a deceptively simple path forward – one that delivers on every promise FedRAMP 20X claims to chase with the proposed automated continuous monitoring, assessment, and reporting: lower costs, reduced time, enhanced autonomy for innovation, and improved risk posture. No need to build a bureaucratic labyrinth. No need to reinvent shared responsibility matrices.

And best of all? No need to write a single page of compliance documentation.

Bold? Sure. But it's at least serious – unlike some of the more performative techno-theatrics that masquerade as reform. While some corners of the market are foaming at the mouth to birth the next unicorn startup solving problems that shouldn't exist in the first place, the rest of us are just trying to defend national systems from actual threats in a tax-payer-responsible fashion.

So, let's zoom out and get real.

The "Why" – Again

Let's reset the mission: Preserve or enhance risk protection while reducing time, cost, and toil by eliminating work that doesn't contribute to production-validated resilience.

Not better paperwork. Not faster compliance theater. Real security.

The "What" – Radical Simplicity via Resilience Testing

Forget checklists. Let's perform actual testing on actual systems.

CSPs open their production environments to live testing by authorized 3PAOs (and optionally AI agents, for those into cybernetic sadomasochism). The 3PAOs act as sanctioned adversaries – running continuous randomized attack simulations against running workloads.

- **Breaches found?** CSPs pay fines to the government, just like emissions penalties in clean air models. A threshold of failures is defined though at which point you lose your ATO altogether.
- **False negatives or collusion?** We bake in rules for audit transparency and "insider trading" prevention. No gaming the system with safe breaches or pre-coordinated testing.
- **Test results?** Pushed to a secure internal system visible to agencies only. Agencies don't get a stack of PDF compliance reports – they get breach reports, remediation plans, and follow-ups tracking whether the fix held or flopped.

In this model, control requirements purely become guidance for building resilient systems, but CSP's are welcome to implement in whatever way they feel confident will work (i.e., innovate). If they are wrong, they'll find out quickly. If they keep being wrong, they are not a trusted provider. We don't care which form of architecture configuration or technology standard you used, whether you have a mountain of policy, process, and procedures documented in 1,000 pages of an SSP, or how pretty the dashboard was in which you reported it. There are no participation trophies.

The "When" – Quarterly, Randomized

Testing cycles occur at least once a quarter, triggered randomly. This ensures CSPs stay ready, rather than dressing up for the known date of an audit like it's senior prom. Thou shall know neither the day nor hour.

The "Where" – Production. Period.

No more hypothetical environments where everything's compliant but nothing's real. Testing should hit the same systems federal users depend on – under the same conditions attackers would exploit.
Worried about downtime or exposure?

CSPs can optionally use perfectly mirrored replica environments (if they really want to spend on the replica), but the keyword is perfect. No hiding behind "representative staging" that shares nothing but a name. Key point is that is the CSP's choice to innovate around.

The "How" – Resilience Engineering

This model leans on the practices already used by the leading security-conscious organizations in the world: resilience engineering, chaos testing, automated red teaming, and post-incident learning loops.

We track what happens when things go wrong – or almost go wrong. Below is a proposed metric framework, grouped by outcome, behavior, and systemic integrity:

Resilience Test Outcome Metrics

Metric	Definition	Why it Matters
Breach Occurrence Rate (BOR)	% of test cycles where a breach was simulated successfully.	Direct measure of environment vulnerability.
Time to Detection (TTD)	Time between breach initiation and detection.	Measures monitoring and alerting efficacy.
Time to Containment (TTC)	Time from detection to isolation of threat.	Reflects incident response maturity.
Time to Remediation (TTR)	Time from containment to tested fix.	Gauges recovery speed.
Re-breach Recurrence Rate (RRR)	% of previously breached systems breached again.	Indicates whether remediations hold.

Operational Resilience Behavior Metrics

Metric	Definition	Why it Matters
Degradation Without Failure (DWF)	# of tests where the system bent but didn't break.	Shows graceful failure behavior.
Auto-Remediation Rate (ARR)	% of incidents mitigated by automation.	Captures maturity of self-healing infra.
False Negative Detection Rate (FNDR)	% of undetected simulated intrusions.	Measures blind spots.
Control Evasion Rate (CER)	# of times test agents bypassed controls.	Reveals weak points in control implementation.

Systemic Integrity & Feedback Loop Metrics

Metric	Definition	Why it Matters
Remediation Completion Rate (RCR)	% of fixes from prior cycles that are complete and validated.	Tracks follow-through and accountability.
Remediation Velocity (RV)	Avg. time from finding → fix → deployment.	Encourages fast, high-quality patching.
Control Drift Rate (CDR)	% of previously validated controls that failed later.	Detects system entropy.
Operational Stability Score (OSS)	Composite of uptime, incident response, and regression.	Sets a baseline resilience benchmark.

Optional Aggregated Metrics:

- **Resilience Index (RI):** Weighted composite score across categories for comparison across CSPs or time.
- **Accountability Delta (AD):** Time lag between breach and validated fix. The smaller the delta, the stronger the loop.

"Why This Works"

Let's review the fundamentals of automation:

1. Don't automate things that shouldn't exist.
2. Only automate things that return real value.

In this model we achieve the following:

1. We eliminate nearly all time and costs needed for compliance documentation development and maintenance, and continuous scanning and reporting (manual or automated). CSPs can continue with these practices internally as desired for operations.
2. We eliminate much time and costs spent on refactoring of application architectures, cloud architectures, and development / administrative operations for no reason other than to say it meets a compliance standard. There will need to be costs and time considered for ensuring a CSP has a resilient architecture, but these are likely a fraction of the exiting bill.

3. We allow CSPs the autonomy to innovate. It's likely there may need to be some lines drawn (e.g., such as systems and personnel being U.S. based or having U.S. citizenship, bill of material and supply chain restrictions, the need to be audited by an accredited 3PAO, etc.). Even these would allow for significantly more autonomy than exists today or through the new proposed automation though.

The current FedRAMP automation trajectory is heading toward a future where we create the OSCAL of Continuous Monitoring – a technically brilliant, exorbitantly expensive solution to a self-inflicted problem. A bureaucracy simulator in JSON.
Instead, what if we focused on outputs that matter?

"Final Thought"

If the goal is real risk reduction at improved cost and time, we don't need a bigger, better compliance machine. We need to replace the machine with something that behaves like reality. And before we go for innovation of automation, we need to consider innovation of thought.

If all else fails though, there's always money in the banana stand.

About Coalfire

The world's leading technology infrastructure providers, SaaS companies, and enterprises – including the top 5 cloud service providers and 8 of the top 10 SaaS businesses – rely on Coalfire to strengthen their security posture and secure their digital transformations. As the largest firm dedicated to cybersecurity, Coalfire delivers a comprehensive suite of advisory and managed services, spanning cyber strategy and risk, cloud security, threat and vulnerability management, application security, privacy, and compliance management. A proven leader in cybersecurity for the past 20 years, Coalfire combines extensive cloud expertise, advanced technology, and innovative approaches that fuel success. For more information, visit **Coalfire.com.**

Copyright © 2025 Coalfire. All rights reserved. The information in this document is subject to change at any time based on revisions to applicable regulations and standards. Any forward looking statements are not predictions and are subject to change without notice. Coalfire is not responsible for any errors or omissions.