A Deterministic, CVSS–Environmental Method for FedRAMP Rev5 VDR/VER Vulnerability Prioritization

Matthew Venne
Chief Technology Officer, stackArmor

June 2026

Abstract

The FedRAMP Vulnerability Detection and Response (VDR) and Vulnerability Evaluation and Reporting (VER) rules define *Potential Agency Impact* (PAIN, levels N1–N5) and a remediation–timeframe matrix, together with evaluation rules for likely exploitability and internet reachability. FedRAMP deliberately does *not* prescribe how to derive a PAIN level from an individual vulnerability on a specific asset. This memo argues that the missing classifier already exists, in standardized form, as the **Environmental metric group of CVSS**: the Confidentiality, Integrity and Availability *Requirements* (CR/IR/AR) and the Modified Impact Sub-Score were designed precisely to re-weight a vulnerability's impact by what is at stake in a given deployment. We present a closed-form, reproducible mapping from FedRAMP's requirements onto CVSS Environmental metrics, derive PAIN and the remediation deadline from it, and give fully worked calculations. We also present an *example* mechanism — "asset archetypes" — for assigning CR/IR/AR systematically, while emphasizing that each Cloud Service Provider (CSP) SHOULD derive its own assignment from its system categorization. A reference architecture and a sample scan complete the memo.

Contents

1 Status of This Memo and Terminology

This document is informational. It does not itself define a FedRAMP requirement; it specifies a standardized *method* for satisfying the existing VDR/VER requirements.[1]

The key words MUST, MUST NOT, SHOULD, SHOULD NOT, and MAY are to be interpreted as described in RFC 2119.

PAIN
Potential Agency Impact, the FedRAMP N1–N5 rating of a finding.

CR/IR/AR
The CVSS Environmental Confidentiality / Integrity / Availability *Requirements* of an asset.

ISC
Impact Sub-Score (here, the CVSS *Modified* Impact Sub-Score).

[1]The VDR and VER rules launched on 2026-06-24 as part of the FedRAMP Consolidated Rules for 2026 and apply to both FedRAMP 20x and Rev5 offerings. This memo retains its original Rev5-oriented title for link stability; the method applies unchanged under both.

LEV / NLEV
Likely Exploitable Vulnerability / Not Likely Exploitable Vulnerability (a VER axis).

IRV / NIRV
Internet-Reachable Vulnerability / Not Internet-reachable Vulnerability (a VER axis).

VEX
Vulnerability Exploitability eXchange, a machine-readable assertion about whether a finding affects a specific product or deployment.

Class
The provider Certification Class (A–D) selecting a remediation table.

2 Motivation: the gap between VDR/VER and the scanner

A vulnerability scanner emits, per finding, a CVE identifier, a qualitative severity, and (often) a CVSS Base vector. FedRAMP Rev5 VDR/VER, by contrast, asks the provider to answer a different question: *what is the potential impact on the agency if this vulnerability is exploited on this asset, and how quickly must it therefore be remediated?* The two are not the same. A "Low"-rated CVE on a crown-jewel datastore can carry more agency impact than a "High"-rated CVE on a throwaway sandbox.

FedRAMP supplies the *output* vocabulary (the PAIN buckets and the remediation matrix) and the *evaluation axes* (likely-exploitable, internet-reachable) but leaves the *classifier* — the function from (CVE, asset) to PAIN — to the provider. An ad-hoc or analyst-by-analyst classifier is neither reproducible nor defensible. This memo supplies a deterministic one.

3 Key insight: FedRAMP's requirements are CVSS Environmental metrics

The central claim of this memo is that FedRAMP's notion of "impact on the agency" maps, almost one-to-one, onto metrics that already exist in the CVSS v3.1 specification's *Environmental* group. The Environmental group exists for exactly this purpose: to let the consuming organization re-score a vulnerability according to the criticality of the affected asset in *its* environment.

FedRAMP VDR/VER concept	**CVSS mechanism that satisfies it**
Asset data sensitivity / FIPS-199 C-I-A categorization	Security Requirements `CR`, `IR`, `AR`
Magnitude of potential agency impact	Modified Impact Sub-Score (`ISC`)
Likely exploitable (VER)	`LEV`: EPSS threshold / CISA KEV catalog membership
Internet reachable (VER)	`IRV`: per-finding payload reachability (companion method)
Single- vs. multi-agency blast radius	scope multiplier on the PAIN tier
Provider Certification Class	selector of the remediation deadline table

Table 1: FedRAMP requirements fall directly onto CVSS Environmental metrics.

The consequence is that a conforming implementation need not invent a bespoke risk algebra. It need only (a) assign each asset its CR/IR/AR, and (b) evaluate the two VER axes. The arithmetic that turns those inputs into a defensible impact magnitude is the published CVSS Environmental formula.

4 The model

We define PAIN as a function of a *severity* term and a *scope* term:

$$\text{PAIN(finding)} \;=\; f\big(\,\text{SEVERITY},\ \text{SCOPE}\,\big).$$

SEVERITY answers "how badly does this vulnerability's impact land on *this* asset?" — the CVE's impact metrics re-weighted by the asset's CR/IR/AR. **SCOPE** answers "how many agencies' data does the asset hold?" — single (1) or multiple (> 1). SEVERITY is mapped to a FedRAMP customer-effect word (Minimal, Narrow, Disruptive, Debilitating); the word and scope together select the N-level.

5 Two threat inputs: EPSS and the CISA KEV catalog

IN PLAIN TERMS

Beyond the CVE's own CVSS vector, the method pulls in exactly two facts about how each flaw behaves in the wild: how *likely* exploitation is (EPSS, a statistical score) and whether exploitation is *known to be happening* (the CISA Known Exploited Vulnerabilities catalog). Either one tightens the remediation *clock*; neither changes the PAIN *level*, which is driven by the CVSS vector and the asset's requirements alone.

Two governed threat inputs feed the exploitability column:

- **EPSS probability** → statistical likelihood. A finding at or above the provider-selected threshold θ_{epss} is Likely Exploitable (LEV) (Section 8).
- **KEV membership** → confirmed exploitation. A CVE listed in the CISA KEV catalog is LEV regardless of its EPSS probability — observed exploitation in the wild overrides any statistical estimate. KEV membership also independently triggers the `VDR-TFR-KEV` remediation dates (Section 12), which run on CISA's published schedule irrespective of the matrix.

How it plays in — three quick illustrations (Class C, single-agency):

- *KEV moves the clock (and starts a second one).* An N3 finding with EPSS 0.30 (below threshold) sits in the slow column: **128 days**. It appears in the KEV catalog: it becomes LEV — the deadline drops to **16 days** (internet-reachable) or 32 days (not) — and the CISA-published due date applies in parallel under `VDR-TFR-KEV`.
- *EPSS moves the clock statistically.* The same N3 finding, not in the KEV catalog but with EPSS 0.85 ($\geq \theta_{\text{epss}} = 0.70$), is LEV on likelihood alone and lands on the same 16/32-day clocks.
- *Neither present.* Sub-threshold EPSS, not in the KEV catalog, and not caught by the `FRD-LEV` unauthenticated-automation floor (Section 8) → the finding is scored purely on its CVSS vector and stays in the slow column at its base tier.

6 The mathematics

IN PLAIN TERMS

This is the engine room — you can skim the equations and still follow the idea. Every flaw can hurt three things: **secrecy**, **correctness**, and **uptime**. For each, we multiply "how badly the flaw breaks it" by "how much this system depends on it," then combine the three into a single damage score between 0 and 1. Finally we translate that score into a plain word. Bigger number = more damage. The only place human judgment enters is the three cut-points that decide where one word becomes the next.

The expected output is a PAIN level. The path is: compute an asset-specific impact scalar S from CVSS impact values and the asset's CR/IR/AR; translate S into a customer-effect word (Minimal, Narrow, Disruptive, or Debilitating); then combine that word with the multi-agency flag to select N1–N5.

The numerical weights and cap in Eq. (1) are taken from the CVSS v3.1 specification. The normalization, word thresholds, and multi-agency mapping are this method's additions; Section 6.1 delineates that boundary precisely.

WHAT THE SYMBOLS MEAN

- C, I, A — how badly *this flaw* breaks secrecy, correctness, and uptime (from the CVE's CVSS vector). None / Low / High become 0 / 0.22 / 0.56.
- $\mathrm{CR}, \mathrm{IR}, \mathrm{AR}$ — how much *this system* needs secrecy, correctness, and uptime (from its archetype, asset value, or other governed asset metadata). Low / Medium / High become 0.5 / 1.0 / 1.5.
- ISC — the three combined into one "damage to this system" number.
- S — that damage rescaled to a clean 0–1 range.
- W — the plain English word S maps to.
- m — does the system serve one agency (0) or many (1)? Resolved hierarchically from governed asset metadata.
- $\theta_1, \theta_2, \theta_3$ — the three cut-points where the word changes.

Impact metric weights. Each of the affected vulnerability's effective Confidentiality, Integrity and Availability impact metrics (C, I, A) takes a weight. The initial values come from the CVSS Base vector; an evidence-gated Modified value can override them as described in Section 10:

$$C, I, A \in \{\, \text{None} = 0, \quad \text{Low} = 0.22, \quad \text{High} = 0.56 \,\}.$$

Security Requirement weights. Each asset's Requirement (CR, IR, AR) takes a multiplier:

$$\mathrm{CR}, \mathrm{IR}, \mathrm{AR} \in \{\, \text{Low} = 0.5, \quad \text{Medium} = 1.0, \quad \text{High} = 1.5 \,\}.$$

Modified Impact Sub-Score. The per-dimension impacts and requirements combine in CVSS's bounded complement-product form: multiply the arithmetic complement on each dimension, then take the complement of that product:

$$\mathrm{ISC} \;=\; \min\Big[\, 1 - \big(1 - C{\cdot}\mathrm{CR}\big)\big(1 - I{\cdot}\mathrm{IR}\big)\big(1 - A{\cdot}\mathrm{AR}\big), \;\; 0.915 \,\Big]. \tag{1}$$

The cap $0.915 = 1 - (1 - 0.56)^3$ is the CVSS ceiling (all-High impact at Medium requirements). We then normalize to a unit scalar:

$$S \;=\; \frac{\mathrm{ISC}}{0.915} \;\in\; [0, 1]. \tag{2}$$

IN PLAIN TERMS

Reading Eqs. (1)–(2) step by step. Each of the three products — $C \cdot \mathrm{CR}$, $I \cdot \mathrm{IR}$, and $A \cdot \mathrm{AR}$ — multiplies two facts: how badly the flaw damages one property (from the CVE's vector) and how much this system depends on that property (from its requirements). A severe flaw on a system that does not care produces a small number; a modest flaw on a system that cares deeply produces a larger one. This is where "same CVE, different asset, different score" happens.

Subtracting each product from 1 flips it, within the arithmetic, from "damage dealt" to "share of that dimension left intact." Multiplying the three complements and subtracting the result from 1 is a bounded complement-product aggregation. It has the same algebraic form as the probability that at least one of several independent events occurs, but the C/I/A weights are not probabilities and the method makes no statistical-independence claim. Unlike simply adding the three products, which could exceed 1, this construction stays bounded and gives each additional affected dimension diminishing marginal weight.

The $\min(\ldots, 0.915)$ cap is CVSS's, not ours. The largest unweighted impact combination is all-High: $1-(1-0.56)^3 = 0.915$. High requirements multiply each dimension by 1.5, so an individual product can reach 0.84, and the three-dimension combination reaches approximately 0.996. The cap prevents that adjusted result from exceeding CVSS's global maximum MISS. The CVE determines which dimensions are affected and their baseline magnitude; the asset requirements determine how strongly those dimensions count. Asset criticality can move a finding toward the scale's worst case, but not beyond its fixed ceiling. That fixed, specification-defined maximum also gives Eq. (2) a stable denominator: $S = 1.0$ means "maximum attainable MISS," and the word thresholds retain the same meaning across assets with different requirement profiles.

Severity word. S is bucketed into a FedRAMP customer-effect word by three cut points $(\theta_1, \theta_2, \theta_3)$:

$$W(S) = \begin{cases} \text{Minimal} & S < \theta_1 \\ \text{Narrow} & \theta_1 \le S < \theta_2 \\ \text{Disruptive} & \theta_2 \le S < \theta_3 \\ \text{Debilitating} & S \ge \theta_3 \end{cases} \qquad (\theta_1, \theta_2, \theta_3) = (0.25,\ 0.55,\ 0.80). \tag{3}$$

The cut points are the model's *one* calibratable judgment. They SHOULD be documented and back-tested against analyst-rated findings, and an implementation MUST treat them as governed configuration rather than an in-band, ad-hoc setting.

Scope and the N-level. Let $m \in \{0, 1\}$ be the asset's multi-agency flag, resolved from governed asset metadata using the most-specific applicable value. Scope is purely hierarchical — an asset is multi-agency only if the asset itself, a containing boundary, or the provider default is tagged multi-agency. The N-level follows from the word and scope:

$$\mathrm{PAIN} = \begin{cases} \mathrm{N1} & W = \text{Minimal} \\ \mathrm{N2} & W = \text{Narrow} \\ \mathrm{N3} & W = \text{Disruptive},\ m = 0 \\ \mathrm{N4} & W = \text{Disruptive},\ m = 1 \ \text{ or } \ W = \text{Debilitating},\ m = 0 \\ \mathrm{N5} & W = \text{Debilitating},\ m = 1 \end{cases} \tag{4}$$

6.1 Where this departs from standard CVSS scoring

Equation (1) is algebraically identical to the CVSS v3.1 Modified Impact Sub-Score (MISS), with $C/I/A$ denoting the effective Modified impact values MC/MI/MA — or their inherited Base values when the Modified metrics are Not Defined. The metric weights, requirement multipliers, complement-product formula, and 0.915 cap are standard CVSS; the abbreviated notation is ours. The departures begin after MISS, with what this method does — and deliberately does not do — with that intermediate.

In the v3.1 specification, MISS feeds the full Environmental Score. For Modified Scope Unchanged, the specification multiplies MISS by 6.42; Modified Scope Changed uses a different impact expression. It then combines Modified Impact with Modified Exploitability, built from `MAV/MAC/MPR/MUI`, applies the Temporal multipliers `E/RL/RC`, caps the score at 10, and performs the prescribed rounding. MISS is not the specified final Environmental Score. We stop at MISS and do not carry any of that downstream machinery into the severity term. Three decisions follow, each with a reason.

We drop the exploitability half of the score. PAIN is Potential Agency Impact: the question is how badly the flaw lands if it fires, not how likely it is to fire. Exploitability is not discarded; it is routed to the axes FedRAMP built for it, the LEV/IRV remediation columns (Section 8), where threat and exposure are measured directly. CVSS's exploitability metrics encode technical preconditions for exploitation; EPSS estimates exploitation probability, and KEV membership records observed exploitation. That separation is consistent with the exposure- and threat-based prioritization of BOD 26-04. Folding `MAV/MAC/MPR/MUI` back into the severity term would let exploitability influence both the row and the column. The seam is clean by construction: impact picks the row; exploitability and reachability pick the column.

We normalize instead of computing the 0–10 blend. CVSS transforms MISS, together with its other inputs, into a standardized 0–10 Environmental Score. Our deliverable is a classification — four words and five N-levels — so the scalar exists only to be bucketed. Equation (2) rescales MISS to the natural axis for drawing those lines: S is the fraction of the maximum attainable MISS. A threshold of $\theta_3 = 0.80$ therefore means "the cut point is 80% of maximum MISS." It is not a claim that the vulnerability causes 80% of some physically measurable damage; the CVSS coefficients are not a ratio-scale measurement of harm. The division changes no ordering — it relabels the axis. The substantive choice is stopping at MISS.

We do not repurpose CVSS Scope. The full Environmental formula branches on the Changed/Unchanged Modified Scope metric. CVSS Scope asks whether exploitation can affect resources governed by a security authority beyond the vulnerable component's authority; it is not limited to sandbox or privilege-boundary escape, and Modified Scope can reflect the local environment. FedRAMP's blast-radius concern here asks a different question: whether the affected asset carries one agency's data or several agencies' data. The multi-agency flag in Eq. (4) carries that concern directly. Because the two concepts are not semantically interchangeable, the CVSS Scope machinery is omitted rather than renamed or repurposed.

The result is not a conformant CVSS Environmental Score. It reuses the standard MISS intermediate, routes exploitability and reachability to FedRAMP's remediation columns, represents multi-agency blast radius explicitly, and normalizes only for classification. A reviewer verifying the CVSS provenance can compare Eq. (1) with the published MISS formula; every choice after it is enumerated above and in Appendix C.

7 Fail-safe behavior

Missing metadata MUST NOT *lower* PAIN. If a system has not been classified yet, the method assumes the worst rather than the best: an asset that carries no classification SHOULD resolve to a deliberately conservative default (e.g. CR/IR/AR all High), which scores loudly and surfaces the asset for classification rather than hiding it. "Unknown" is treated as "serious until proven otherwise" — silence never makes a problem look smaller than it is.

8 Remediation: the VDR-TFR-PVR matrix

IN PLAIN TERMS

The N-level says *how bad*; this step says *how fast*. Two real-world facts tighten the clock: is the flaw actually being exploited (or very likely to be), and is the system reachable from the internet? More severe + more exploitable + more exposed = less time to fix. A lookup table turns those into a concrete deadline, stricter for higher-assurance providers.

The remediation deadline is selected by three values: the provider Certification Class, the PAIN level, and an exploitability *column* derived from the two VER axes:

$$\text{deadline} = M\big[\,\text{Class}\,\big]\big[\,\text{PAIN}\,\big]\big[\,\text{column}\,\big], \qquad \text{column} \in \{\,\text{LEV+IRV},\ \text{LEV+NIRV},\ \text{NLEV}\,\}.$$

where, for a provider-selected EPSS threshold θ_{epss} (this memo uses 0.70):

$$\begin{aligned} \text{LEV} &= \big(\text{EPSS} \geq \theta_{\text{epss}}\big) \ \vee\ \big(\text{CVE} \in \text{KEV}\big) \ \vee\ \big(\text{IRV}_{\text{direct}} \wedge \text{unauthenticated automation}\big), \\ \text{IRV} &= \text{the } \textit{finding} \text{ is internet-reachable (evaluated per vulnerability, not per asset).} \end{aligned}$$

The third LEV disjunct is not a provider option: it implements the `FRD-LEV` floor note that "any vulnerability that an automated unauthenticated system can exploit over the internet is a likely exploitable vulnerability." A deterministic proxy suffices: the finding is IRV via *direct* exposure and its CVSS vector permits unauthenticated automation (`AV:N/PR:N/UI:N`).

DEFINITION

Internet reachability is a property of the *(vulnerability, asset) pair*, not of the asset alone. Per `FRD-IRV`, an internet-reachable vulnerability is one that "might be exploited or otherwise triggered by a payload originating from a source on the public internet" — explicitly including resources with *no direct internet route* that process payloads received indirectly; `VER-EVA-EIR`'s canonical example is SQL injection reaching a database on a private network through the application tier. The companion paper distinguishes the directly computable network branch from the transitive content branch. Its current operational profile is:

$$\text{IRV}_0(v,a) = \mathbb{1}[N(a) \cap X_N(v) \neq \varnothing] \ \vee\ \left(\mathbb{1}[J(v) = \text{confirmed}] \cdot C_0(a) \cdot (1 - P_0(v,a))\right),$$

where $N(a)$ is the directly exposed network surface, $X_N(v)$ is the direct network surface required by the vulnerability, $J(v)$ is the reusable CVE-level indirect-trigger promotion, and $C_0(a)$ is the current coarse assertion that the component receives internet-derived content. $P_0(v,a)$ is a current, assessment- and ConMon-backed assertion that the relevant trigger or input-control class is prevented before vulnerable processing. Recurring DAST and remediation refresh that assertion; stale, out-of-scope, failed, or contradicted evidence sets P_0 to zero. CVSS `AV:L/A/P` excludes a finding from the direct branch but does not prevent promotion into the indirect branch. The paper recommends this operational profile under today's metadata conditions, under which reachability converges operationally toward direct accessibility plus known uncovered indirect triggers. A separate, non-normative full-information form shows how structured content trigger profiles and optional application-edge evidence can refine

the result when publishers or provider-selected tools supply them. A separate, bypass-free IAP/VPN remote-access boundary can remove the backend's direct path; application-level authentication is not an IRV/NIRV gate and instead informs LEV or attested mitigation. It lowers PAIN only when its enforced scope demonstrably reduces customer effect after successful exploitation. The full equations, optional extension model, and worked cases are specified in the companion paper *Internet Reachability at Scale under the FedRAMP VDR/VER Rules*.

Restricting the threat inputs to EPSS and the KEV catalog — and to nothing else — is itself a design opinion (see Appendix C). The day-valued tables are given in Appendix B. PAIN-1 carries no FedRAMP deadline.

9 Worked examples

IN PLAIN TERMS

The same formula, run by hand on realistic cases. Watch the *same* vulnerability land at very different N-levels depending only on which system it sits on — that is the entire point of the method.

We use C, I, A for impacts and CR/IR/AR for requirements; all arithmetic follows Eqs. (1)–(4). For reference, the severity word used in each example is assigned by:

$$W(S) \;=\; \begin{cases} \text{Minimal} & S < \theta_1 \\ \text{Narrow} & \theta_1 \le S < \theta_2 \\ \text{Disruptive} & \theta_2 \le S < \theta_3 \\ \text{Debilitating} & S \ge \theta_3 \end{cases} \qquad (\theta_1, \theta_2, \theta_3) = (0.25,\ 0.55,\ 0.80).$$

Example 1 — RCE on a crown-jewel datastore, multi-agency. $C = I = A = 0.56$ (High), CR=IR=AR=1.5 (High), $m = 1$.

$$\begin{aligned} \text{ISC} &= \min\big[\,1 - (1 - 0.84)^3,\ 0.915\,\big] = \min[\,0.9959,\ 0.915\,] = 0.915, \\ S &= 0.915/0.915 = 1.000 \ \Rightarrow\ W = \text{Debilitating}, \\ m = 1 \ &\Rightarrow\ \boxed{\text{N5}}. \end{aligned}$$

Example 2 — the same CVE on a sandbox, single-agency. $C = I = A = 0.56$, CR=IR=AR=0.5 (Low), $m = 0$.

$$\begin{aligned} \text{ISC} &= 1 - (1 - 0.28)^3 = 1 - 0.72^3 = 0.6268, \\ S &= 0.6268/0.915 = 0.685 \ \Rightarrow\ \text{Disruptive} \ \Rightarrow\ \boxed{\text{N3}}. \end{aligned}$$

Identical vulnerability, opposite ends of the matrix — the asset half of the score (CR/IR/AR) is doing the work.

Example 3 — availability-only DoS on a message backbone. A CVE with $C = 0$, $I = 0$, $A = 0.56$ (e.g. a parser DoS), on an asset with CR=IR=AR=1.5, $m = 0$. EPSS = 0.76. The vulnerable parser sits on the inter-broker replication surface, which receives no internet-derived

data, so the finding is not internet-reachable (surface-excluded under the companion model — had the flaw been in *message-payload* parsing it would be transitively IRV).

$$\begin{aligned} \text{ISC} &= 1-(1-0)(1-0)(1-0.84) = 1-0.16 = 0.84, \\ S &= 0.84/0.915 = 0.918 \;\Rightarrow\; W = \text{Debilitating},\; m = 0 \;\Rightarrow\; \boxed{\text{N4}}. \end{aligned}$$

The single dimension $A \cdot \text{AR} = 0.84$ alone clears the Debilitating bar. Remediation: LEV (EPSS ≥ 0.70) $\wedge$ NIRV; at Class C this is M[C][N4][LEV+NIRV] = 8 days. Note this finding may carry a *Low* vendor severity — PAIN is decoupled from the qualitative label.

Example 4 — information disclosure on a metadata-only service. $C = 0.56,\ I = 0,\ A = 0$, on an asset with CR=0.5 (Low), IR=AR=1.5.

$$\begin{aligned} \text{ISC} &= 1-(1-0.28)(1)(1) = 0.28, \\ S &= 0.28/0.915 = 0.306 \;\Rightarrow\; W = \text{Narrow} \;\Rightarrow\; \boxed{\text{N2}}. \end{aligned}$$

A confidentiality leak on a service that holds only operational metadata is reconnaissance, not a debilitating breach — captured by the low CR.

10 Mitigation and the PAIN Ladder

IN PLAIN TERMS

FedRAMP does not demand that every flaw be *fixed* inside its deadline — it demands that the potential damage come *down the ladder* on schedule. There are two ways to climb down: make the flaw hurt less if it fires (its *row* drops to a lower N-level), or make it harder to reach or exploit (its *column* moves to a slower clock). Every step down must be backed by evidence somebody signed, every step is recorded, and no step may claim the damage is exactly zero — "zero" is a different kind of claim with its own paperwork (VEX).

FedRAMP's response model is a ladder, not a binary. `VER-RPT-VDT` requires reporting the "time and Potential Agency Impact N-rating of each completed and evaluated reduction in Potential Agency Impact N-rating" and the "estimated time and target Potential Agency Impact N-rating of next reduction in Potential Agency Impact N-rating" — the rule *expects* PAIN to be re-estimated downward as mitigations land — and the `VDR-TFR-PVR` timeframes are satisfied by partial mitigation to a lower N-rating (Appendix B). A deterministic classifier therefore needs a deterministic *lever*: a governed way to recompute PAIN when a mitigation changes the facts.

The lever is the other half of CVSS Environmental. Section 6 uses one half of the Environmental group, the Security Requirements (CR/IR/AR). The other half is the *Modified Base* metrics, which the CVSS specification provides precisely for capturing mitigations, configurations, and compensating controls present in a specific environment. The two halves split cleanly along the model's own seam:

- **Asset side — static.** CR/IR/AR and the multi-agency flag describe what the asset *is*. They never move to make a deadline.
- **Vulnerability side — dynamic.** The Modified impact metrics (`MC/MI/MA`) describe what the flaw can still do *here*, given the controls actually deployed. They are evidence-gated and move as mitigations land.

Recomputation runs through the same arithmetic — substitute the modified values for C, I, A in Eq. (1) — and the word, the N-level, and the deadline follow exactly as before. No new formula, no side channel.

Concrete row-movers. These are ordinary hardening actions, not magic wands; each has a defensible causal story for the dimension it lowers:

- Drop the vulnerable process to a non-root user with a read-only filesystem — the effective Integrity (and often Confidentiality) of an RCE falls.
- Scope the application's database credential to a single schema — a SQL injection's C/I falls: the payload still arrives, but it reaches less.
- Disable the vulnerable feature — the impact of a code path that cannot be invoked falls. (Removing the vulnerable code *outright* goes further and graduates to a VEX disposition; see below.)
- Confine egress — default-deny outbound rules break exploit chains that need a second stage.
- Rate-limit the affected endpoint — Availability impact falls for volume-dependent flaws.

High availability: automatic where the exploit cannot chase it. Verified redundancy is a legitimate Modified-Availability (`MA`) evidence class, and its facts are machine-checkable up front: replica count, *required* zone anti-affinity (a `preferred` rule is a scheduling hint, not a guarantee — verify the actual pod spread), a PodDisruptionBudget, and health-checked automatic restart. Where those facts hold, the credit applies *automatically*: `MA` High→Low, recomputed through Eq. (1), with the detected facts as the recorded evidence. The one carve-out is scoped by the companion reachability method: a finding that is *internet-reachable* does not take the automatic step, because IRV is precisely the statement that an attacker can deliver the trigger — and re-deliver it. Replication defends against *uncorrelated* failures; every replica in every zone runs the identical vulnerable code, and the load balancer obligingly delivers a repeated exploit to all of them. For that corner the step is still available, but with a per-finding citation stating why the trigger cannot be repeated at line rate (expensive to construct, rate-limited, state-dependent). Everywhere else — the majority of findings, where no internet path delivers the trigger — nothing is hammering the replicas, and redundancy covers exactly the accidental and one-shot failures it was built for. Adopting the automatic rule is a governed-configuration choice signed once (Appendix C), with the residual on record: an attacker already inside the boundary can still repeat a trigger against an NIRV finding. There is still no graduated zone schedule: the availability impact metric takes the values High, Low, and None, None is reserved for the VEX disposition path, so the ladder has exactly one rung here regardless of how many zones the deployment spans.

Worked example: Log4Shell egress-deny. In December 2021, most organizations' first effective Log4Shell mitigation was not a WAF. One of the most widely deployed was default-deny egress — a NetworkPolicy or security-group change blocking outbound LDAP/RMI from application subnets. Log4Shell's RCE requires a second-stage fetch of the malicious class from the attacker's LDAP server, so blocking egress breaks the chain; but DNS-based exfiltration of environment variables (`${env:AWS_SECRET_ACCESS_KEY}`) remained observed in the wild — which is exactly why the Modified Confidentiality drops to Low and never to None. The arithmetic runs through Eqs. (1)–(4) unchanged. Base, with $C = I = A = 0.56$ (High) on a CR=IR=AR=1.5 (High) multi-agency asset:

$$\mathrm{ISC} = \min\big[\,1 - (1 - 0.84)^3,\ 0.915\,\big] = 0.915, \quad S = 1.000 \ \Rightarrow\ \text{Debilitating},\ m = 1 \ \Rightarrow\ \text{N5},$$

on the Class C LEV+IRV clock of **2 days** (Log4Shell is LEV twice over: KEV-listed, and caught by the `FRD-LEV` floor). With the evidenced egress-deny, MC = MI = MA = 0.22 (Low):

$$\text{ISC} = 1 - (1 - 0.33)^3 = 0.699, \quad S = 0.699/0.915 = 0.764 \;\Rightarrow\; \text{Disruptive},\; m = 1 \;\Rightarrow\; \boxed{\text{N4}}.$$

The N5 timeframe is satisfied by the reduction, the reduction is reported under `VER-RPT-VDT`, and work proceeds on the N4 clock. The evidence is the NetworkPolicy object snapshot and enforcement evidence showing the egress path removed; optional flow or graph evidence can help bind that control to the workload. One clock is untouched: Log4Shell is a KEV, so the `VDR-TFR-KEV` remediation date runs independently of any reduction.

Guardrails. The lever only ever lowers apparent risk, so it carries the same governance as the disposition overlay of Section 14:

1. A modification MAY only *lower* a dimension on cited, signed evidence, and MUST be re-evaluated whenever the control it relies on changes. The evidence bar scales with the PAIN being reduced.
2. A modification MAY lower High to Low but MUST NOT lower any dimension to None. Claiming *zero* impact is an applicability claim, and it travels as a VEX disposition (`not_affected`), not as a metric modification.
3. The audit trail records base → modified per dimension with the evidence references, mirroring the retained-PAIN pattern: the pre-mitigation rating stays on record and the reduction is attributable and reversible.

The rest of the menu, from the same incidents. The guardrails are not hypothetical — each has already played out in public:

- **Code removal (Log4Shell).** Apache's own interim recommendation removes the vulnerable class from the jar entirely:

  ```
  zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class
  ```

  In this model that claim travels as a VEX disposition (`vulnerable_code_not_present`) evidenced by the modified jar hash — not as a Modified-impact claim (guardrail 2).
- **Config kill-switch, and the re-evaluation lesson (Log4Shell).** The JVM flag `-Dlog4j2.formatMsgNoLookups=true` was initially treated as a complete mitigation — until CVE-2021-45046 showed non-default configurations (Thread Context Map patterns) where lookups still fired, and the flag was downgraded to insufficient. That is guardrail 1 playing out in the real world: mitigations carry timestamps and are re-evaluated when the facts change; a one-time silent suppression would have outlived its justification.
- **regreSSHion (CVE-2024-6387).** The published mitigation is one line of `sshd_config`: `LoginGraceTime 0`, closing the pre-authentication RCE race at the cost of a connection-exhaustion DoS risk — a mitigation that *trades* impact dimensions (Modified C/I fall, Modified A is retained or worsens), and one no WAF could ever address.
- **Spring4Shell (CVE-2022-22965).** Spring's interim workaround was a `@ControllerAdvice` restricting the data binder's `disallowedFields` (`"class.*"` and friends) — an application-configuration mitigation shipped while framework upgrades were scheduled.

Column moves: the other axis. Reachability and exploitability mitigations do not touch the impact metrics at all — they move the finding's *column*. Network segmentation that verifiably cuts the payload path, or a current prevention assertion under the companion model, can flip IRV to NIRV; a qualifying remote-access boundary (the companion model's Gate 3) defeats the

FRD-LEV unauthenticated-automation floor. VDR-TFR-PVR is satisfied by landing on *any* longer clock, whether the step down was a row or a column. The textual hook for the column half is FRD-PMV/FRD-FMV: FedRAMP defines mitigation over *likelihood or* PAIN, and an evidence-backed reachability or exploitability reduction is the likelihood half.

Where the ladder ends. The descent is bounded, not endless. It terminates in one of four states: *remediation*; *full mitigation* (FRD-FMV: the vulnerability "will still exist (with negligible risk) until it has been remediated," the remaining remediation handled in routine operations per VDR-TFR-RMN, KEVs excepted); *PAIN-1*, which carries no timeframe (Appendix B) and likewise returns the finding to routine operations; or *formal acceptance* — VER-TFR-MAV requires categorizing any vulnerability that is not or will not be fully mitigated or remediated within 192 days of evaluation as an accepted vulnerability, tracked on the accepted track of Section 14. It is no coincidence that 192 days is also the largest cell in the timeframe matrix: the grid and the acceptance boundary meet by construction. Nor does each rung demand a fresh mitigation. Each step lands on a strictly longer clock (at Class C, LEV+IRV: $2 \rightarrow 4 \rightarrow 16 \rightarrow 48$ days), so the common pattern is a single mitigation that buys schedule, followed by remediation on the longer clock — or, where remediation is genuinely impossible, acceptance at the 192-day boundary rather than an indefinite treadmill.

WAF realism. A WAF virtual patch is one item on this menu, not the presumed answer. VER-EVA-EIR calls payload interception "the simplest way" to prevent exploitation — not the expected way. Narrow, CVE-specific virtual patches are often practical (blocking Log4Shell's JNDI lookup strings); generic rules — say, parameterized SQL-injection signatures across a real API surface — carry false-positive risk to legitimate customer traffic, and the claimed mitigation MUST be re-validated whenever the rule set changes. The incident record above bears this out: under real deadline pressure, the mitigations that actually shipped were configuration flags, code removal, privilege and egress confinement, and binding restrictions — surgical, testable, and zero risk to customer traffic — while the WAF rule was the fallback for the workloads that could not be touched (appliances, vendor binaries). That is precisely its position in this method. The companion reachability paper develops the corresponding posture recommendation: keep the finding IRV and attest the perimeter mitigation via signed VEX rather than reclassifying reachability on WAF evidence.

11 Reference architectures and sample scans

11.1 Single-agency, Class C

Figure 1 shows a representative deployment with each component tagged by archetype. Table 2 shows the corresponding per-finding output for a fictional Class C, single-agency offering. The PAIN column is computed by the method above; the Classical Severity column is the scanner's qualitative rating, shown to make the decoupling explicit.

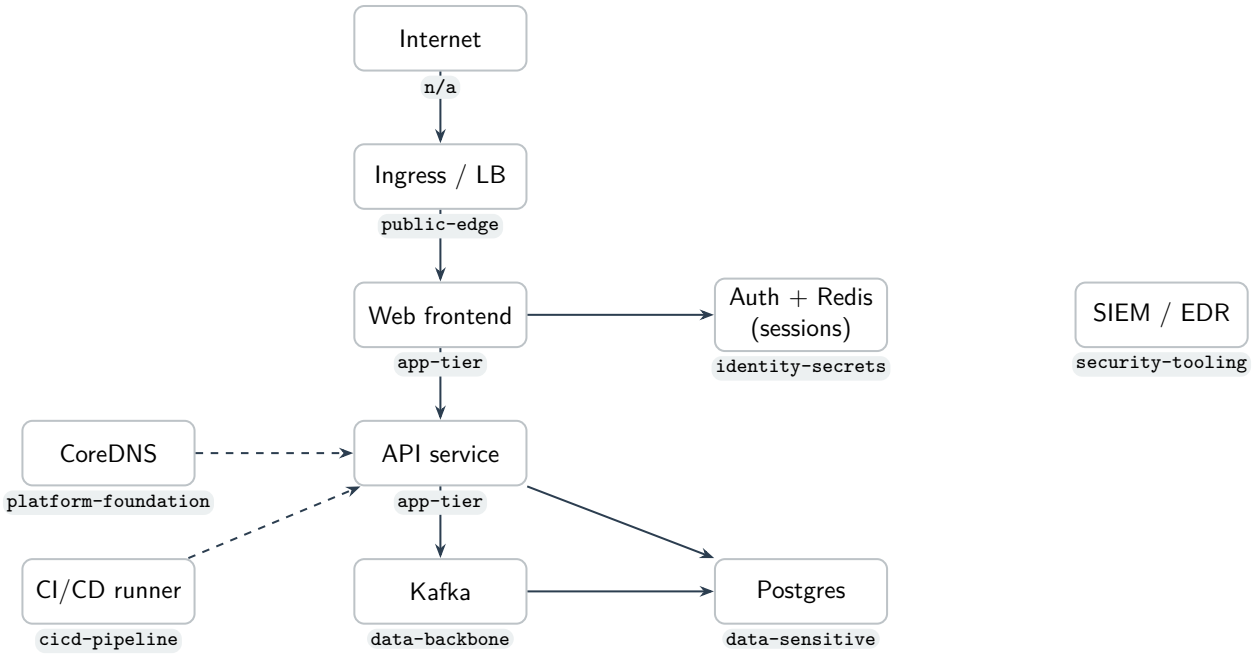

Figure 1: Reference architecture; each node carries its `asset-archetype` tag.

CVE	Resource (archetype)	Classical	EPSS	IRV	PAIN	Remediation
CVE-2026-xxx1	ingress (`public-edge`)	CRITICAL	0.97	yes	N4	4 days
CVE-2026-xxx2	payments-db (`data-sensitive`)	CRITICAL	0.94	yes (t)	N4	4 days
CVE-2026-xxx3	redis (`identity-secrets`)	HIGH	0.50	no	N4	64 days
CVE-2026-xxx4	kafka (`data-backbone`)	LOW	0.76	no	N4	8 days
CVE-2026-xxx5	web (`app-tier`)	MEDIUM	0.20	yes	N3	128 days
CVE-2026-xxx6	coredns (`platform-foundation`)	MEDIUM	0.08	no	N2	192 days

Table 2: Sample VDR scan output (fictional data, placeholder CVE IDs; Class C, single-agency). PAIN is computed from the archetype's CR/IR/AR and the CVE impact vector; the deadline is M[C][PAIN][column]. The IRV column is evaluated *per finding* via the companion finding-level model; (t) marks transitive payload exposure. CVE-2026-xxx2 is FedRAMP's canonical case: a content-triggered (SQL-injection-class) finding on a datastore that is not internet-accessible is still IRV because this sample assumes no current P_0 prevention assertion, tightening the deadline to 4 days. CVE-2026-xxx4 — a *Low* vendor severity that still yields N4 (availability-High DoS on a High-availability-requirement asset; Example 3) — stays NIRV because its vulnerable parser sits on the inter-broker replication surface, which receives no internet-derived data.

11.2 Multi-agency, Class D

Figure 2 is a shared, multi-tenant platform serving *several* agencies at the highest assurance (Class D). Its shared components carry more than one agency's data, so they are tagged multi-agency (here via the cluster default, with single-agency tenant workloads tagged as the exceptions). Table 3 shows the result: the same families of flaws now remediate in *hours*.

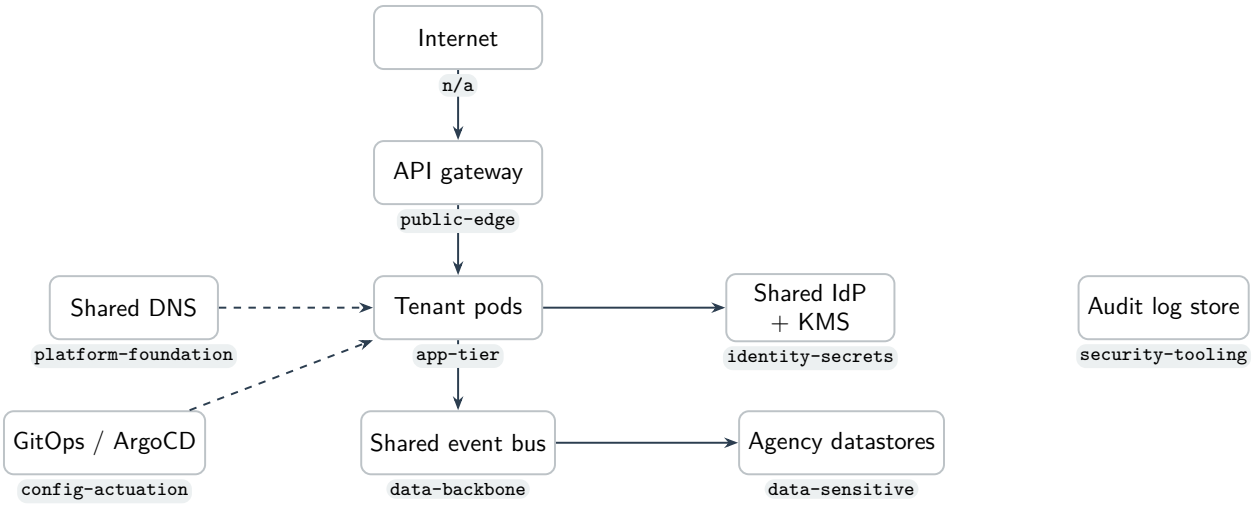

Figure 2: Multi-agency reference architecture; shared components serve several agencies and are tagged multi-agency.

CVE	Resource (archetype)	Classical	EPSS	IRV	PAIN	Remediation
CVE-2026-yyy1	gateway (`public-edge`)	CRITICAL	0.90	yes	N5	12 hours
CVE-2026-yyy2	shared-idp (`identity-secrets`)	CRITICAL	0.95	no	N5	1 day
CVE-2026-yyy3	event-bus (`data-backbone`)	LOW	0.80	yes (t)	N5	12 hours
CVE-2026-yyy4	agency-db (`data-sensitive`)	HIGH	0.40	no	N5	8 days
CVE-2026-yyy5	tenant-pod (`app-tier`)	MEDIUM	0.10	no	N3	64 days
CVE-2026-yyy6	shared-dns (`platform-foundation`)	MEDIUM	0.05	no	N2	192 days

Table 3: Sample VDR scan output (fictional data, placeholder CVE IDs; Class D, multi-agency). Multi-agency scope escalates Disruptive/Debilitating findings, and Class D carries the tightest deadlines — an internet-facing exploitable flaw on the shared edge remediates in **12 hours** (CVE-2026-yyy1). CVE-2026-yyy3 shows a *Low* vendor severity at N5 also remediating in **12 hours**: a content-triggered DoS in message-payload parsing on the shared backbone is transitively internet-reachable (t) through the tenant pods, even though the bus itself is not internet-accessible; the sample assumes no current P_0 assertion covers that trigger. A confidentiality leak on shared DNS (CVE-2026-yyy6) stays N2 and NIRV — "Narrow" does not escalate with scope.

12 Relationship to CISA BOD 26-04

The Rev5 VDR/VER rule is FedRAMP's implementation of CISA Binding Operational Directive 26-04.[2] FedRAMP adopts the directive's *philosophy* — move remediation prioritization off raw CVSS base scores and onto exploitability and mission impact — but substitutes its own instruments for the directive's pillars:

[2]CISA, *Binding Operational Directive 26-04*, 2026-06-10. The VDR rule lists it as the mandating authority, with optional adoption from 2026-07-04, obtain/maintain from 2026-12-07, and grace ending 2027-03-07.

BOD 26-04 pillar	FedRAMP VDR/VER instrument	Status
SSVC drives prioritization	**PAIN** (Potential Agency Impact, N1–N5)	SSVC not named; PAIN replaces it
VEX communicates exploitability	**LEV** (exploitation) + disposition (§14)	VEX not named; role split across LEV and VER disposition
CSAF machine-readable advisories	VER JSON reporting (§14)	format unspecified; CycloneDX VEX proposed (companion)
KEV catalog due dates	`VDR-TFR-KEV` (defers to CISA dates)	retained intact; also consumed as a LEV input
(none)	**IRV/NIRV** and Certification **Class**	added as timeframe axes

Table 4: FedRAMP keeps BOD 26-04's intent but swaps SSVC → PAIN and VEX → LEV+disposition, retains KEV, and adds reachability and Class.

The first row is the deviation that shapes this memo. Because FedRAMP chose *PAIN* rather than SSVC, an SSVC-based prioritizer does not by itself satisfy the rule — and FedRAMP publishes PAIN's *output* buckets and the timeframe table but not the *classifier* that produces an N-level. That gap is exactly what Section 6 fills, which is why this method is expressed in CVSS Environmental terms (the missing input was PAIN, not an exploitability score) rather than in SSVC. The directive's other instruments retain narrower roles: this memo consumes the *KEV catalog* directly as one input to LEV (Section 8) — arguably a cleaner story than substituting for it, since BOD 26-04's own exploitation instrument is kept intact rather than replaced by a derived signal — while VEX carries the post-detection *disposition* of each finding (Section 14). Neither SSVC nor VEX is mandated by the rule by name.

13 Assigning CR/IR/AR: archetypes (an example, not a standard)

IN PLAIN TERMS

Rather than hand-rate every server, we sort systems into a short list of *archetypes* — a database, a login service, a cache, a message bus, and so on — and each archetype comes with its criticality pre-filled. Tag a system with its archetype and the scoring is automatic. Our list is only an *example*: every provider should tailor its own to the data it actually holds.

The model in Section 6 is only as good as the CR/IR/AR assignment that feeds it. Without an archetype approach, a provider would have to tag each asset with CR/IR/AR scores individually. While certainly justifiable and valid, this requires evaluating each asset in isolation. In doing so, we lose a systematic "chain of thought" — for instance, why is a specific host scored CR:L, IR:H, AR:L?

By assigning requirements to archetypes instead, the underlying rationale remains readily understandable and easily auditable. Tagging an asset with its archetype inherits that clear chain of thought. We propose the archetype taxonomy herein as a valid option, while recognizing that not every asset will fit cleanly. CSPs MAY define custom archetypes or adjust the scoring of existing ones at their discretion to map their estate accurately. The goal is to identify the most critical assets as defined by FedRAMP and deprioritize less critical ones.

A simpler, equally valid implementation is to tag assets directly with an `asset-value` instead of an `asset-archetype`. Under that approach, `H` or `High` maps to static `CR:H/IR:H/AR:H`; `M`, `Medium`, or `Moderate` maps to `CR:M/IR:M/AR:M`; and `L` or `Low` maps to `CR:L/IR:L/AR:L`. This schema is easy to implement and audit, and it may be sufficient for smaller or more homogeneous estates. Its tradeoff is that it loses some granularity and the explanatory chain of thought that an archetype catalog provides: every security requirement moves together rather than separately modeling confidentiality, integrity, and availability needs.

> *The archetype catalog in Appendix A is illustrative. A conforming implementation* MUST *derive its CR/IR/AR assignment from its own system categorization (FIPS-199 impact levels, data inventory, authorization boundary). It* MAY *adopt the example catalog as a starting point, but each CSP* SHOULD *own and justify its mapping. Two providers with different data inventories can legitimately assign the same software different requirements.*

The classification rule we recommend is *control-plane lens first*: if an asset can deploy, orchestrate, hold cross-estate credentials, or actuate configuration, classify it by that control function regardless of the data it stores; otherwise classify by the data it holds. The same physical software lands in different archetypes by role — an in-memory store used as a cache is `app-tier`, the same software used as a session/token store is `identity-secrets`, and used as a job broker is `data-backbone`.

For Kubernetes estates, the multi-agency flag can be resolved most-specific-first: workload label, then namespace label, then cluster default. Other deployment models should apply the same rule to their own asset, boundary, and provider-default metadata.

14 Applicability and the disposition via Vulnerability Exploitability eXchange (VEX)

IN PLAIN TERMS

Scoring says how bad a finding is and how fast to fix it. But a scanner's hit is not yet a confirmed problem: the flagged code may not be present, may never run, may need a configuration nobody set, or may already be neutralized by a control. Deciding which is the *disposition* step — and it can only happen *after* a finding is detected, because you cannot pre-judge factors specific to a vulnerability you have not yet seen. This section places that step in the method and names the kind of artifact that records it: VEX.

The PAIN derivation in Sections 6 and 8 is deterministic: given the asset's CR/IR/AR, the VER axes, and the configured thresholds, the N-level and deadline follow by formula. Disposition answers a different question: whether the scanner finding is actually exploitable in this running environment. That requires evidence about the specific vulnerability and deployment — for example, whether the affected code is present, reachable, configured, or already neutralized by a compensating control. The analysis may be automated, but it happens after detection because those facts are finding-specific.

Where FedRAMP requires it. VER requires three things relevant here: evaluate whether a scanner finding is real, record the per-finding disposition, and track vulnerabilities that are accepted rather than remediated. The FedRAMP clauses behind those duties are `VER-EVA-EFP`, `VER-EVA-AIA`, `VER-RPT-VDT`, and `VER-TFR-MAV`/`VER-RPT-AVI`. VER makes the disposition step

mandatory, but it does not prescribe a carrier format. The *Vulnerability Exploitability eXchange* (VEX) is a standardized, machine-readable, signable artifact for carrying those assertions.

Two outcomes. A disposition resolves a finding onto one of two tracks:

- **Not truly affected** — suppressed with a machine-readable *justification* (component or vulnerable code not present, code not reachable, exploitation requires a configuration that is not set, or a compensating control is already in place).
- **Affected but remediation will be late** — under VER's tight clocks a provider MAY be unable to patch before the deadline; the finding is recorded as *accepted*, with an attested mitigation and a planned response.

A disposition overlay, not a re-score. Disposition does not change the arithmetic. Every detected finding is scored as in Section 6; a VEX assertion then moves the finding from the active remediation queue into a suppressed or accepted set *while its computed PAIN is retained*. Keeping the would-be PAIN on record makes every suppression auditable and reversible — the safeguard the Security Considerations (Section 16) call for: a downgrade by disposition is visible, attributable, and can be revisited if the assertion is later contradicted.

A distinct axis. Applicability (VEX) is orthogonal to exploitation-likelihood (LEV, Section 8). LEV asks *how likely is this vulnerability to be exploited in the wild*; VEX asks *is this product, as deployed, affected at all*. A finding can be likely-exploitable in general yet `not_affected` here because the vulnerable path is unreachable. The two are evaluated independently and MUST NOT be collapsed into one.

Guardrails. Because a disposition only ever *lowers* apparent risk — and the more so once the determination is automated — the governing controls matter as much as the analysis. Absent sufficient evidence a finding MUST remain affected, honoring `VER-EVA-AIA`; and the evidence required to suppress SHOULD scale with the retained PAIN, so a KEV-class, internet-reachable, high-N finding demands far stronger corroboration — and human review — than a low-N internal one. Every disposition SHOULD cite the evidence it rests on and be signed, so it is reproducible and attributable; and it SHOULD carry a timestamp and be re-evaluated when the image, configuration, or control it relied on changes, so a suppression cannot outlive the condition that justified it.

15 Determinism, governance, and defensibility

- **Determinism.** Given the same (CVE vector, archetype, VER axes), the method yields the same PAIN and deadline on every run and for every analyst.
- **Two human inputs only.** An asset needs exactly two tags (`asset-archetype`, `multi-agency`); everything else is derived.
- **Governed calibration.** The word thresholds (Eq. (3)) and the EPSS threshold are the only tunable knobs and MUST live in governed configuration, not in ad-hoc cluster state.
- **Auditability.** Each finding can carry the full derivation: the resolved archetype and its source, CR/IR/AR, S, the word, the scope, and the selected matrix cell.

16 Security Considerations

The method trusts the type-tags on each system: if someone mislabels a critical system as trivial, its flaws will look smaller than they really are. Its integrity therefore depends on the trustworthiness of two inputs — the asset archetype tags and the VER axes (EPSS / KEV membership / reachability). Tag spoofing that *lowers* an archetype is a downgrade attack; the fail-safe default and governed calibration mitigate accidental under-classification, but tag assignment SHOULD be controlled like any other security setting. Finally, the method does not detect vulnerabilities; it prioritizes the output of a scanner and inherits that scanner's coverage and accuracy.

17 Conclusion

FedRAMP Rev5 VDR/VER specifies the vocabulary and the axes of vulnerability prioritization but leaves the classifier unspecified. That classifier need not be invented: the CVSS Environmental metric group already encodes "re-weight impact by what is at stake," which is precisely the agency-impact question. By assigning each asset its CR/IR/AR — via archetypes or any other systematic scheme the CSP owns — and evaluating the two VER axes, a provider obtains a deterministic, auditable, and standards-grounded PAIN and remediation deadline for every finding. We offer the example archetype catalog not as a standard but as an existence proof; we encourage each CSP to publish its own.

A companion proposal, *A Finer Exploitability Gradation (VLEV) for FedRAMP VDR/VER*, separately explores tightening the exploitability axis. It is kept out of this memo deliberately: this memo's scope is meeting FedRAMP's requirements as they stand today, not changing them. A second companion, *A CycloneDX VEX Profile for FedRAMP Rev5 VDR/VER Disposition and Response*, specifies the machine-readable format for the disposition step of Section 14.

A Example archetype catalog

Illustrative only (see Section 13).

Archetype	Lens	CR	IR	AR	Typical members
`cicd-pipeline`	control	H	H	H	build/deploy runners, artifact signing, registries
`orchestrator`	control	H	H	H	control plane, etcd, scheduler, coordination
`config-actuation`	control	H	H	H	IaC/GitOps, schema registry, admin/migration
`identity-secrets`	control	H	H	H	IdP/SSO, KMS, secrets managers, session stores
`security-tooling`	control	H	H	M	scanners, SIEM, EDR, runtime security
`change-record`	control	M	M	M	ITSM/ticketing (record only)
`platform-foundation`	control	L	H	H	DNS, NTP, service discovery, L4 internal LBs (metadata only)
`data-sensitive`	data	H	H	H	PII/CUI datastores
`data-backbone`	data	H	H	H	payload queues and brokers, the system-of-record DB
`telemetry-backbone`	data	M	M	M	metrics/trace pipelines, telemetry queues, event buses carrying no agency payload
`app-tier`	data	M	M	M	stateless services, APIs, UIs, caches
`batch-analytics`	data	M	M	L	ETL, reporting, analytics jobs
`public-edge`	data	L	L	H	load balancers, public web, ingress
`internal-tooling`	data	L	L	L	dashboards, metrics/log agents
`dev-test`	data	L	L	L	non-production
`unclassified`	—	H	H	H	fail-safe default for untagged assets

Two assignments deserve their reasoning on the record. First, `data-backbone` is reserved for components that carry agency payload data; a bus that moves only metrics, traces, and heartbeats is `telemetry-backbone`, one grade lower on every dimension. Routing payload data through a telemetry bus is a misclassification finding, not a reason to keep every bus at High. Second, the availability grades split deliberately between the edge and the application tier. `public-edge` keeps AR High because a CVE-grade denial of service hits every replica at once—redundancy defends against hardware failure, not against a flaw shared by the whole class—and an edge-class outage closes the front door for every user. `app-tier` sits at Medium because its blast radius is one service degrading. The same availability-High CVE therefore scores Debilitating on the edge and Disruptive on the application tier, which matches how an operations team actually experiences the two events.

B Remediation deadline matrices

Days; 0.5 = 12 hours. Columns in order LEV+IRV / LEV+NIRV / NLEV. PAIN-1 has no FedRAMP deadline (any class). The day counts below are reproduced *verbatim* from FedRAMP's published **VDR-TFR-PVR** table;[3] Classes A and B share one schedule. The only provider-side choice here is which Certification Class the offering commits to — not the numbers.

[3]FedRAMP, *Vulnerability Detection and Response (VDR)*, Rev5 provider rule, requirement `VDR-TFR-PVR`. Generated markdown pinned at FedRAMP/2026-markdown@2a75592 (commit of 2026-06-24); accessed 2026-06-27. Mandated by CISA BOD 26-04; FedRAMP may revise the values.

	Class A / B			Class C			Class D		
PAIN	L+I	L+N	NLEV	L+I	L+N	NLEV	L+I	L+N	NLEV
N5	4	8	32	2	4	16	0.5	1	8
N4	8	32	64	4	8	64	2	8	32
N3	32	64	192	16	32	128	8	16	64
N2	96	160	192	48	128	192	24	96	192

Note: Values are days measured from *completed evaluation* (`VDR-TFR-PVR` runs its timeframes from evaluation, not detection) within which the finding SHOULD be remediated, fully mitigated, or at least partially mitigated to a lower PAIN rating; 0.5 represents 12 hours.

C Explicit design opinions (calibratable choices)

Everything in this section is a *judgment call* we made — not something FedRAMP or CVSS dictates. We collect them in one place so reviewers can challenge each one, and so any adopter knows exactly which dials are theirs to turn. None of these is load-bearing for the *method*; they are the parameters the method runs on. Each item below names the value we use, why, and the fact that an implementation MAY change it.

1. **PAIN word thresholds** $(\theta_1, \theta_2, \theta_3) = (0.25, 0.55, 0.80)$**.** The cut points that turn the impact scalar S into Minimal/Narrow/Disruptive/Debilitating (Eq. 3). This is the single most consequential opinion — it sets where a finding crosses an N-level boundary. The values should reproduce senior-analyst triage on a back-tested sample; an adopter SHOULD re-calibrate against its own rated findings and MUST treat the result as governed configuration.

2. **EPSS Likely-Exploitable threshold** $\theta_{\text{epss}} = 0.70$**.** The EPSS probability at or above which a finding is treated as LEV (Section 8). Higher means fewer findings on the fast track; FedRAMP leaves the framework to the provider.

3. **LEV is a union, not a blend.** A finding is Likely Exploitable if EPSS $\geq \theta_{\text{epss}}$, *or* it is listed in the CISA KEV catalog, *or* it is internet-reachable via direct exposure and its vector permits unauthenticated automation (`AV:N/PR:N/UI:N`) — whichever fires, with no weighting among them. The third disjunct is not provider discretion: it implements the `FRD-LEV` floor note that "any vulnerability that an automated unauthenticated system can exploit over the internet is a likely exploitable vulnerability."

4. **Impact-only severity, normalized by** 0.915**.** We take CVSS's *Modified Impact Sub-Score* and rescale it to $[0, 1]$ rather than computing the full Environmental Score; Section 6.1 enumerates this choice and every downstream CVSS component the method omits.

5. **CISA Vulnrichment is deliberately excluded.** Two CISA Vulnrichment SSVC values are natural candidate inputs: the exploitation state (as a LEV input) and the technical impact (as an impact floor). We adopt neither. The exploitation role is served by KEV membership — the instrument BOD 26-04 itself names — and a technical-impact floor is rejected because an opaque, externally-fixed impact floor cannot coexist with the evidence-based Modified C/I/A mitigation lever of Section 10: when a fixed floor and cited local evidence disagree, there is no principled precedence between them, and the Environmental layer already re-weights impact per deployment. EPSS and the KEV catalog remain the two governed threat inputs. An adopter MAY reintroduce enrichment sources, at its own risk of exactly that conflict.

6. **Modifications lower High to Low, never to None; evidence scales with the PAIN reduced.** The mitigation lever of Section 10 lets cited, signed evidence lower a Modified impact dimension, but never to None: a claim of *zero* impact is an applicability claim and must travel as a VEX disposition (Section 14), where it is separately governed, auditable, and reversible. And the evidence bar scales with the stakes — lowering an N5 to an N4 demands stronger corroboration than lowering an N2 to an N1, mirroring the disposition guardrails. Both rules are ours, not FedRAMP's; an adopter tightening or relaxing them SHOULD do so in governed configuration.

7. **Automatic high-availability credit is scoped by reachability.** Verified redundancy (replicas, required zone anti-affinity with observed spread, PodDisruptionBudget, automatic restart) applies `MA` High→Low automatically for findings that are *not* internet-reachable; internet-reachable findings take the step only with a per-finding citation explaining why the trigger cannot be repeated at line rate. Rationale: IRV means the attacker can re-deliver the exploit, and replication defends against uncorrelated failure, not a flaw every replica shares. An adopter MAY tighten this (citation everywhere) or, at its own risk, relax the reachability carve-out; either choice belongs in governed configuration with the residual documented.

8. **Scope is purely hierarchical.** An asset is multi-agency only if the asset itself, a containing boundary, or the provider default is tagged so (most-specific wins); there is no automatic per-archetype escalation. (The tempting shortcut — auto-escalating "shared-infrastructure" archetypes to multi-agency — is rejected in favor of explicit, operator-controlled tagging.)

9. **"Archetype", not "type".** We classify assets by their *role/usage pattern*, not their intrinsic kind, and the label reflects that: the same software lands in different archetypes by how it is used (an in-memory store is `app-tier` as a cache, `identity-secrets` as a session store, `data-backbone` as a broker). "Type" would wrongly suggest an intrinsic property of the software and collides with overloaded terms (e.g. Kubernetes Service `type`); "archetype" names the pattern an asset matches in the estate. An adopter MAY rename the label, but the role-not-kind semantics should be preserved.

10. **Fail-safe defaults to conservative High.** An unclassified asset defaults to CR/IR/AR all High so it scores loudly, rather than to a low default. "Unknown" is treated as serious until classified.

11. **CVSS v3.1, not v4.0.** We build on v3.1's closed-form environmental formula because it provides a Modified Impact Sub-Score (Eq. 1) with explicit CR/IR/AR multipliers. CVSS v4.0 retains CR/IR/AR but feeds them into a lookup-based MacroVector rather than the v3.1 formula, and it splits impact into Vulnerable-System (`VC/VI/VA`) and Subsequent-System (`SC/SI/SA`) metrics. A v4.0-only finding can be approximated by mapping `VC/VI/VA` onto $C/I/A$, at the cost of ignoring Subsequent-System impact; while NVD remains v3.1-dominant, an implementation SHOULD prefer the v3.1 vector when both are available.

12. **The archetype catalog and its CR/IR/AR values (Appendix A).** Illustrative only — each CSP SHOULD derive its own from its system categorization. A 3PAO MAY take on the role of validating these mappings during an assessment; however, the CSP remains ultimately responsible if an asset is undertagged, a vulnerability is not remediated in time, and it is subsequently exploited (just as in the commercial world).

13. **The remediation deadline values (Appendix B) are *not* ours to calibrate.** They are reproduced verbatim from FedRAMP's published `VDR-TFR-PVR` table (mandated by CISA

BOD 26-04; see Section 12). The only provider-side choice is which Certification Class (A–D) the offering commits to, selecting the column block; the day counts themselves are fixed by FedRAMP. Listed here only to mark the boundary between what we chose and what the rule dictates.

D Document version history

This memo is a living document. Table 5 tracks its version history.

Version	Date	Author	Summary of Changes
1.0	June 28, 2026	M. Venne	Initial publication.
1.1	June 29, 2026	M. Venne	Various readability updates.
1.2	July 9, 2026	M. Venne	Added the optional `asset-value` assignment scheme as a simpler alternative to `asset-archetype`.
1.3	July 10, 2026	M. Venne	Expanded the mathematical walkthrough of Eqs. (1)–(2) and delineated departures from standard CVSS v3.1 scoring.

Table 5: Document version history.

Acknowledgements

While this paper has a single official author, the underlying work, refinement, and practical validation were a collaborative effort. It would not have been possible without the generous help, critical feedback, and dedicated support of various other team members at stackArmor who helped make it possible.

Normative sources

[VDR] FedRAMP, *Vulnerability Detection and Response (VDR)*, Rev5. Requirements cited: `VDR-TFR-PVR`, `VDR-TFR-KEV`, `VDR-TFR-PDD/PCD/PSD`. Pinned permalink: FedRAMP/2026-markdown@2a75592 (commit of 2026-06-24; generated from upstream `fedramp/rules`). Accessed 2026-06-27.

[VER] FedRAMP, *Vulnerability Evaluation and Reporting (VER)*, Rev5. Requirements cited: `VER-EVA-EIR`, `VER-EVA-EFP`, `VER-EVA-AIA`, `VER-RPT-VDT`, `VER-TFR-MAV`, `VER-RPT-AVI`. Pinned permalink: FedRAMP/2026-markdown@2a75592 (commit of 2026-06-24). Accessed 2026-07-01.

[DEF] FedRAMP, *FedRAMP Definitions*. Definitions cited: `FRD-IRV`, `FRD-LEV`. Pinned permalink: FedRAMP/2026-markdown@2a75592 (commit of 2026-06-24). Accessed 2026-07-01.

[EGR] M. Venne, *Internet Reachability at Scale under the FedRAMP VDR/VER Rules*, stackArmor, July 2026. Companion paper specifying the reachability determination used here.

[BOD] CISA, *Binding Operational Directive 26-04*, 2026-06-10.

[KEV] CISA, *Known Exploited Vulnerabilities (KEV) Catalog.* LEV input; `VDR-TFR-KEV` schedule.

[CVSS] FIRST, *CVSS v3.1 Specification* — Environmental metric group and Modified Impact Sub-Score.

[EPSS] FIRST, *Exploit Prediction Scoring System.*