// ARTICLE · PUBLISHED 4 NOVEMBER 2026

FedRAMP and GovRAMP: Evidence as a Command

Both programs judge how software is built and tested, so the controls went into the rules every AI agent reads and the machines that run every build

Jason Newell · MAX Research Collective · Article ART-CLD-012 · Published 4 November 2026
Licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0), https://creativecommons.org/licenses/by/4.0/
Original: https://www.maxresearchcollective.com/articles/fedramp-and-govramp.html
Cite as: Newell, J. (2026). FedRAMP and GovRAMP: Evidence as a Command. MAX Research Collective, ART-CLD-012. https://www.maxresearchcollective.com/articles/fedramp-and-govramp.html

Building to FedRAMP and GovRAMP with an AI workflow

FedRAMP and GovRAMP decide whether government can buy your software. Both judge how you build and test it, as well as what it does. We built those rules into an AI development workflow. Every change our agents make is built and tested inside a boundary we control and leaves a record; the evidence is a command anyone can run.

What FedRAMP and GovRAMP are

Both are security authorizations for cloud software sold to government. FedRAMP covers federal agencies; GovRAMP covers states, local governments and other public bodies, schools among them. Each certifies that a named system, inside a stated boundary, meets a fixed set of security controls and keeps meeting them; a product's features sit outside what either one checks.

Table 1. FedRAMP and GovRAMP side by side.

	FEDRAMP	GOVRAMP (FORMERLY STATERAMP)
Full name	Federal Risk and Authorization Management Program	GovRAMP, the operating name of StateRAMP since its 2025 rebrand
Who runs it	The General Services Administration (GSA) and its FedRAMP office	A nonprofit, StateRAMP Inc., doing business as GovRAMP
Who it serves	Federal agencies buying cloud services that hold unclassified federal data	State and local governments and public education; modeled on FedRAMP
Legal standing	Began in 2011 under an Office of Management and Budget (OMB) policy memo; made law by the FedRAMP Authorization Act, signed 23 Dec 2022 in the FY2023 National Defense Authorization Act (NDAA)	A membership program that states adopt through their own procurement rules
Control set	NIST SP 800-53 Rev. 5, chosen into baselines: Low and Low-Impact SaaS (LI-SaaS) 156 controls, Moderate 323, High 410	The same NIST SP 800-53 catalog, in baselines aligned with FedRAMP's
Who assesses	A FedRAMP-recognized third-party assessment organization (3PAO), then an agency authorizes	A 3PAO, then GovRAMP's program office verifies
Reuse	An agency must presume another agency's FedRAMP authorization is adequate (the 2022 Act)	One assessment serves every member jurisdiction that requires it

NIST SP 800-53, the control catalog the National Institute of Standards and Technology publishes, is the shared language. It's a list of security and privacy controls (access control, configuration management, supply chain, testing, audit); each baseline selects from it. FedRAMP is also moving to a faster track, FedRAMP

20x, with automated checks built on evidence — the FedRAMP Marketplace lists 20x certifications beside the classic ones.

Why they matter, and when they are required

An authorization is a condition of sale. A federal agency may not put its data in a cloud service that lacks a FedRAMP authorization at the right level. A growing number of states write GovRAMP, or its equivalent, into their contracts. Without the authorization the deal doesn't happen, however good the software is.

- When it's required. FedRAMP applies when a cloud service stores, processes or transmits unclassified federal information for an agency. GovRAMP applies when a participating state or local government requires it of its vendors.
- The level follows the data. Low covers public or low-risk data, Moderate covers most government systems; High covers law enforcement and emergency services, and financial or health data whose loss would be severe.
- Health care sits close to High. Systems for patient records, for benefits and for public health hold data whose disclosure harms people. They draw Moderate or High baselines on top of HIPAA, the federal health privacy law.

The reason behind them is the supply chain. An attacker who changes how software is built reaches every customer at once, so the frameworks look past the running product to who can change the code, where it's built and tested, what it depends on, and whether the release that ships is the one that was tested. In an AI workflow the agents write and test the code, so they and the machines they use are part of the system being authorized.

What they ask of building and testing

The build pipeline must sit inside a boundary the provider controls, or every outside service in it must be named and justified. It must also leave evidence an assessor can check. These are the NIST SP 800-53 controls that land on it, with the ISO/IEC 27001:2022 clauses we tracked beside them (ISO 27001 is the international standard for an information security management system).

Table 2. The controls that land on a build pipeline.

AREA	NIST 800-53 CONTROLS	ISO 27001 ANNEX A	WHAT IT MEANS FOR A PIPELINE
Where it is built	SA-9, SA-10, CM-2	A.8.30, A.8.31	Builds run inside the boundary; an external builder is named and justified
Baseline	CM-2, CM-6, SA-10	A.8.9	The build environment is written down and pinned
Change control	CM-3, CM-5	A.8.32	Only reviewed, tested changes reach a release
Supply chain	SR-3, SR-4, SR-11	A.5.21	Third-party code and actions pinned; a bill of materials per release
Integrity	SI-7	A.8.25	The release shipped is the one built and tested, bound by hashes
Least privilege	AC-6, CM-7	A.8.2	Build jobs and tokens get only what they need
Testing	SA-11	A.8.29	Every change is tested before it merges
Vulnerabilities	RA-5, SI-2	A.8.8	Dependencies scanned; flaws fixed within set times
Audit	AU-2, AU-12	A.8.15	Each build and release leaves a record

How we built it into an AI workflow

Our software is written mostly by AI agents: Claude Code sessions, with Codex, Gemini and Copilot as reviewers. Muther, a desktop app that runs the Collective's build and test machines, coordinates them. Compliance went in at the start. It's part of the rules every agent reads before it writes a line, and part of the machinery that runs each build.

1. Rules every agent reads first. Each repository has one entry file (AGENTS.md) that every agent loads, pointing to binding rules, a ship checklist and a ledger of failures already paid for. An agent that skips a step is caught by the checks. The class of mistake goes into the ledger so it isn't repeated.
2. Tests and checks run on machines we control. Every pull request is tested on self-hosted runners on the Collective's PC rather than a shared cloud; jobs that need a secret never run there.
3. Builds inside the boundary, in Docker. Release installers are built on the PC in a single-use container. It has a pinned base image, a cap on cores and memory, no capabilities, no secrets, and the source as a read-only snapshot of the exact commit. Muther builds her own releases this way and builds MAX4's in a second container; MAX4's own build lane takes over only if she doesn't answer.
4. Evidence as a command. compliance-check prints one line per control (actions pinned to commit hashes, least-privilege tokens, lockfile-only installs, the dependency audit, release integrity) and fails the build on any gap. Every release carries a CycloneDX software bill of materials (SBOM) and a manifest naming where it was built, the tool versions and the lockfile hash.
5. Nothing merges red, nothing ships vulnerable. Continuous integration (CI) fails on any high or critical dependency advisory; the AI releases only what passed CI on the main branch.
6. Visible and audited. Each build container is labelled with what it builds and who started it, shown on Muther's runners display, and every start and stop, and every clean-up, goes to an audit log. Each entry says how to undo it.

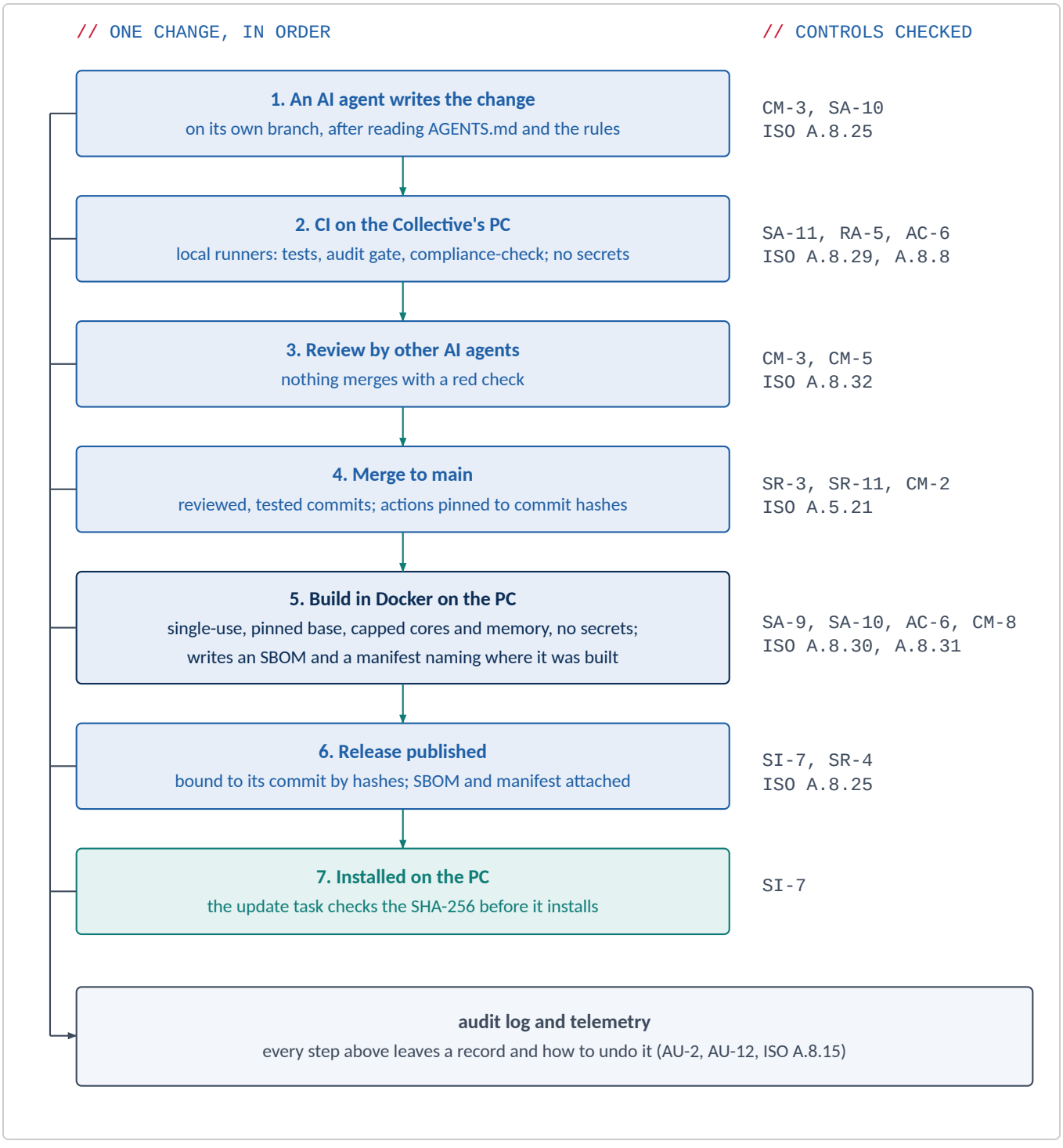


Figure 1. Follow one change down the page: each step names the controls it answers, and every step writes to the audit log at the foot.

The build in Docker (step 5 in the figure) is what moved building inside the Collective's boundary. Every other step was already local or hosted by GitHub, which stays a named outside service.

What is proven today, and what is still open

The build and test side is in place and checked by command. No authorization has been sought yet, because the product isn’t offered to an agency; we assume the Moderate baseline.

Table 3. Where each control stands on 25 October 2026.

CONTROL	STATE	EVIDENCE
Actions pinned to commit hashes	Met	compliance-check actions-pinned
Least-privilege tokens in every workflow	Met	token-least-privilege
Installs from the lockfile only	Met	locked-dependencies
Builds inside the boundary (PC, Docker)	Met	each release's manifest says pc-docker
Build isolated, pinned base, no secrets	Met	tests on the container's arguments
SBOM and build provenance per release	Met	sbom.cdx.json, manifest.json
High and critical advisories block CI	Met	audit gate; seven vulnerable packages fixed on 25 Oct 2026
Every check required before merge	Open	branch protection, the Collective's step
Installer code signing	Open	needs a hardware-held key; the Collective's decision
Fix deadlines for moderate advisories	Open	FedRAMP Moderate allows 90 days
Supplier register for outside services	Open	GitHub, the AI services and the code reviewers each need a line

The bar stayed where it was, and the evidence grew: each agent leaves a pull request, a test run and an audit line. The same check an assessor would run is the one that blocks a bad merge.

Why this has to continue: MARA

This work is the ground floor for MARA, the Medical Agentic Resident Assistant. MARA is meant to hold one patient's whole record, across a lifetime, for every role in their care. The patient, providers, imaging, the lab, pharmacy, treatment teams, the payer and billing each read and add to the same story, instead of rebuilding it from scattered charts. She learns from outcomes, across a person and across similar patients. Every recommendation she makes comes with its reason, and a person makes the decision.

For a clinic, that means the facts from one visit are there at the next, a patient isn't asked for the same history at every handoff, and a warning can come from what happened to patients like this one. A system trusted with that has to meet HIPAA. Wherever public health and government programs depend on it, it has to meet FedRAMP or GovRAMP too. So these controls carry forward. MAX4, the generation before her, already builds and tests to them — the security and regulatory evidence MARA needs exists before any patient depends on her.

Sources

- FedRAMP: the program site and Marketplace, including FedRAMP 20x (read 25 Oct 2026).
  - fedramp.gov
- FedRAMP Codified: a new law aims to streamline federal security authorizations, Davis Wright Tremaine, Jan 2023: the 2022 Act, the presumption of adequacy, the 2011 OMB start.
  - dwt.com · fedramp authorization cloud services
- FedRAMP Revision 5 explained, Cloud Security Alliance, 14 Jul 2023: baseline control counts.
  - cloudsecurityalliance.org · fedramp revision 5 explained
- NIST SP 800-53 Rev. 5: Security and Privacy Controls for Information Systems and Organizations.
  - csrc.nist.gov · final

- StateRAMP is now GovRAMP, Government Technology, 14 Feb 2025.
  - govtech.com · stateramp is now govramp reflecting broader participation
- Our own record: docs/COMPLIANCE.md and scripts/compliance-check.mjs (AlienInterface), docs/compliance/BUILD_AND_TEST_FROM_MUTHER.md and docs/reference/architecture/Building_MARA.md (MAX4).

Glossary

TERM	DEFINITION
3PAO (third-party assessment organization)	an independent assessor FedRAMP recognizes. Its assessment is what an agency or GovRAMP's office relies on.
Baseline	the set of NIST controls chosen for one impact level, Low, Moderate or High. The level follows how much harm the data's loss would do.
CI (continuous integration)	the checks that run on every proposed change before it merges. Here they run on the Collective's own runners.
CycloneDX	an open format for a software bill of materials. Every release here carries one.
FedRAMP	the federal program that authorizes cloud services for federal agencies. Without it an agency can't put its data in the service.
FedRAMP 20x	FedRAMP's faster track, built on automated, evidence-based checks. Its certifications sit beside the classic ones in the Marketplace.
GovRAMP	the state and local counterpart to FedRAMP, run by a nonprofit and called StateRAMP until 2025. One assessment serves every member jurisdiction.
HIPAA	the federal health privacy law. Health systems sold to government meet it on top of their FedRAMP or GovRAMP baseline.
ISO/IEC 27001	the international standard for an information security management system. Its Annex A clauses were tracked beside the NIST controls.
LI-SaaS (Low-Impact SaaS)	a FedRAMP baseline for low-risk software sold as a service. It shares the Low control count.

TERM	DEFINITION
MARA (Medical Agentic Resident Assistant)	the Collective's design for a clinical assistant that recommends while a person decides. She's why these controls have to carry forward.
MAX4	the desktop generation of the MAX AI resident, from October 2026 on. It already builds and tests to these controls.
Muther	the desktop app that runs the Collective's build and test machines and builds releases in Docker.
NDAA (National Defense Authorization Act)	the annual federal defence law. The FY2023 act carried the FedRAMP Authorization Act.
NIST SP 800-53	the catalog of security and privacy controls from the National Institute of Standards and Technology. Both programs select their baselines from it.
OMB (Office of Management and Budget)	the White House office whose 2011 policy memo started FedRAMP.
SBOM (software bill of materials)	a list of every component a release contains. It lets an assessor check the supply chain.
Self-hosted runner	a machine the Collective owns that runs CI jobs, in place of a shared cloud runner.

CLOUD / DEVOPS ART-CLD-012

// ABOUT THE AUTHOR

Jason Newell leads organizations and people in technology, and is a researcher, deep technologist and builder — from a CRAY to agentic AI. He founded MAX Research Collective to build AI for regulated industries that runs where its owner has control, with healthcare as its strongest focus. About Jason → Connect on LinkedIn →

Newell, J. (2026). MAX Research Collective, ART-CLD-012. CC BY 4.0