EZRMF · WHITEPAPER · OVERVIEW

Automated ATO with AI Agents

How ezRMF compresses the DoD RMF lifecycle from months to weeks using sandboxed agents, retrieval over evidence, and one-click eMASS exports.

SHE BASH LLC · **EZRMF** 2026 · V4.2 DISTRIBUTION: UNCLASSIFIED

The problem with manual RMF

> Every system seeking Authority to Operate produces the same artifacts — System Security Plan, control implementation narratives, evidence, POA&M, eMASS imports — and every team produces them the same way: copy-paste between Word, Excel, SharePoint, and an evidence folder no one trusts. The result is months of work that does not improve the security posture of the system.

An RMF package for a medium-sized DoD application typically requires:

- 500–900 NIST 800-53 controls assessed across implementation, test, and remediation.
- Thousands of CCIs (Control Correlation Identifiers) mapped to those controls and linked to evidence.
- 3,000–8,000 evidence artifacts (configuration exports, screenshots, scan results, interview notes).
- An eMASS import package containing implementation status, test results, POA&M, hardware/software inventory, and SSP narratives.

The work is not difficult. It is voluminous, repetitive, and trivially error-prone — the worst possible profile for human labor and the best possible profile for narrow, well-scoped AI agents.

What ezRMF replaces

MANUAL WORKFLOW	EZRMF EQUIVALENT	TIME SAVED
Type control narratives into spreadsheets	Bulk import from existing SSPs / SARs; AI extraction	50–70%
Maintain hardware/software inventory by hand	Scheduled AWS auto-discovery (EC2, RDS, S3, Lambda, EKS, ALB, VPC, SG, EBS)	~80%
Copy Security Hub findings into POA&M line items	Auto-generated POA&M entries with NIST control mapping	~90%

MANUAL WORKFLOW	EZRMF EQUIVALENT	TIME SAVED
Hunt evidence by filename or folder location	Semantic vector search across every uploaded artifact	Qualitatively unlimited
Assemble eMASS import package by hand	One-click .xlsx export for test results, POA&M, and inventory	~100%

How ezRMF works

ARCHITECTURE

ezRMF is a two-container application — frontend + sandboxed AI agent runtime — built for AWS GovCloud (IL-5) but deployable anywhere a container can run. It is intentionally infrastructure-light so it can be stood up forward, in disconnected environments, or alongside an existing eMASS instance.

Four automation pillars

1. Connect your cloud

Scheduled discovery agents enumerate AWS resources on a cron and write them straight into the project inventory with provenance metadata (account, region, last-seen). Security Hub findings auto-create POA&M items, mapped to NIST controls via CCI overlap.

2. Import everything

Existing SSPs, SARs, STIG CKLs, vulnerability scans, and policy documents are ingested via PDF / DOCX / XLSX parsers. Controls and CCIs are extracted in bulk and merged into the project baseline, leaving the human-in-the-loop free to review rather than re-type.

3. AI does the work

Seventeen sandboxed agent skills handle the repetitive jobs: assessing controls, organizing evidence, drafting narratives, linking findings to CCIs, generating diagrams, and proposing remediation paths. Every agent action is logged to an immutable audit trail; nothing is autonomous in a way that hides reasoning from the analyst.

4. Export to eMASS

One-click exports produce the three .xlsx files needed for an eMASS import package — Test Result Import Template, POA&M, and Hardware/Software Inventory — formatted to the latest published schema, with CUI banners, prepared metadata blocks, and per-row provenance.

> **Net effect:** a system that would normally absorb 3–6 months of analyst time can have a defensible draft package in 4–8 weeks, with the analyst spending their effort on decisions rather than data entry.

CSRMC alignment

ezRMF is purpose-built around the DoD Cyber Survivability Risk Management Construct (CSRMC) five-phase lifecycle. Each phase has explicit tooling support — see *CSRMC 5-Phase Alignment Reference* for the detailed coverage matrix.

PHASE	ACTIVITY	EZRMF SUPPORT
Design	System categorization, control selection	FIPS-199 wizard, baseline import, tailoring
Build	Implementation, evidence capture	BoE pipeline, CCI mapping, suggestion queue
Test	Assessment, test result recording	Assess workbench, AP coaching, CKL ingest
Onboard	Dashboard / DevSecOps integration	Continuous telemetry, audit log, MCP tools
Operate	Continuous monitoring, cATO	Auto-discovery cron, Security Hub bridge

Defensibility

"AI-generated compliance" is reasonably treated with skepticism. ezRMF mitigates the obvious failure modes:

- **Sandboxed agents.** Agent skills run in an isolated bubblewrap sandbox with a constrained network and filesystem. They cannot exfiltrate evidence or modify the host.
- **Full audit trail.** Every CCI link, every narrative edit, every export carries an author (human or agent) and a timestamp. Reviewers see exactly which sentences were drafted by an agent.
- **Approve-before-canonical.** Agent suggestions land in a queue. Nothing reaches the SSP, the POA&M, or an eMASS export without an analyst's approve action.
- **Reproducible exports.** Every export is derived from the database at export time. There is no "stale Word document drift" between what ezRMF says and what eMASS receives.

Security posture

ezRMF ships as FIPS-validated Chainguard containers with zero known CVEs at release. STIG hardening is applied at build time; the running pod meets the ASD V6R4 Application Security and Development STIG. OIDC handles authentication; role-based access controls restrict which users can sign, approve, and export — independent of any LLM token a user might possess.

Outcomes

Two customer engagements (anonymized) ran ezRMF end-to-end on a medium-sized cloud-native application:

ACTIVITY	MANUAL BASELINE	WITH EZRMF	REDUCTION
Initial inventory + baseline	~3 weeks	~2 days	~85%
Implementation narratives (400+ controls)	~5 weeks	~9 days	~70%
Evidence collection + CCI linking	~6 weeks	~2 weeks	~66%
POA&M generation	~1 week	~1 day	~80%
eMASS import package assembly	~4 days	~30 minutes	~99%

What ezRMF does *not* do

ezRMF does not replace the AO, the ISSM, or the ISSO. It is not a substitute for security engineering judgment. It does not auto-sign packages, auto-mitigate findings, or auto-approve risk. It does the data work so the humans can do the security work.

Contact **info@shebash.io** for a demonstration or a copy of the platform technical specification.