Ritma Pilot Proposal

CISO / CTO Briefing

Prepared by Umesh (Ritma)

Version: Pilot-ready, December 2025

Ritma is aimed at teams that already have logging, monitoring, and policy enforcement in place, but who need to be able to *prove*, to themselves, their board, auditors, and regulators, that these systems behaved as claimed at specific points in time.

Ritma: Cryptographic Evidence Fabric Around Your Existing Stack

- Ritma adds a *cryptographically provable audit fabric* around your existing SIEM / IAM / policy stack, without forcing you to replace tools that already work.
- Pilot is sidecar-only: we observe and prove, but do not sit inline on critical production paths or become a new availability risk.
- You get hash-chained logs, truth snapshots ("Git commits for reality"), and verifiable evidence packages that can be checked independently from the Ritma team.
- Commercially: a 10-day free guided demo, then an optional paid pilot with clear success criteria, decision gate, and no lock-in if it does not meet your bar.

1 Chapter 1: Problem Why Evidence Is Broken Today

Modern organizations already collect vast quantities of logs and metrics, but when regulators or incident response teams ask hard questions, those logs are often not enough to give a confident, defensible answer. This chapter explains why the status quo around evidence is fragile.

- Logs are cheap to generate and easy to silently lose or rewrite.
- Compliance evidence is manual, brittle, and rarely cryptographically provable.
- Policy and access decisions are hard to reconstruct and defend months later.
- Evidence pipelines from raw logs to final audit decks are often undocumented, making it hard to show auditors exactly how results were produced.
- Responsibility for end-to-end evidence integrity is scattered across teams; no single system is accountable for the full chain of custody.

2 Chapter 2: Architecture, Data Flows, and Evidence Model

This chapter describes how Ritma is deployed in a pilot, what components are involved, and what data flows through the system. The goal is to make it obvious where Ritma sits in relation to your existing SIEM, IAM, and policy stack.

- Components: `utld` daemon (truth layer) and `utl_cli` (CLI client/reporting).
- Storage: `dig_index.sqlite`, `decision_events.jsonl`, `compliance_index.jsonl`.
- All data remains on your infrastructure; payloads can be redacted or hashed.
- Integration pattern: applications and policy engines send structured events to Ritma over a local socket; Ritma never needs direct access to production databases or secrets.
- Network stance: in a pilot, Ritma does not require outbound internet access; all state is local so you can test it entirely inside your own security boundary.

3 Chapter 3: Security Posture, Threat Model, and Failure Modes

From a security and risk perspective, the key questions are: what happens if a Ritma node is compromised, if its storage is corrupted, or if keys leak? This chapter outlines our assumptions and how integrity and blast radius are managed in each scenario.

- We assume baseline OS/IAM controls; Ritma focuses on integrity and provability of evidence.
- Node compromise or DB corruption is detectable via hash-chains and truth snapshot verification.
- Pilot mode is non-inline: if Ritma is down, production continues; you only lose new evidence, not availability.
- Ritma's own operations are logged and can be included in evidence packages, so you can see when policies were changed, when evidence was exported, and by whom.
- Signing and verification keys are configurable to use your existing secret-management or KMS setup in a production deployment.

4 Chapter 4: Pilot Offer 10-Day Demo to Paid Pilot

This chapter sets out the commercial and operational path: a low-friction 10-day demo, followed by an optional, tightly scoped paid pilot with clear success criteria and an explicit yes/no decision point.

- 10-day free demo on staging/non-prod with 1–2 real workflows.
- If useful: 60–90 day paid pilot with defined control objectives and SLOs for evidence.
- Optional co-enhancement track for reports, integrations, and deeper proofs.
- Clear roles: we expect a CISO/CTO sponsor, a security engineer, and an application owner to be named for the pilot so decisions are fast and grounded.
- Commercial terms, data boundaries, and success criteria are written down up front so there are no surprises at the end of the pilot.

5 Chapter 5: Evidence, Evaluation, and Next Steps

Finally, we outline how you can independently verify Ritma's outputs, how you should evaluate the pilot, and what concrete steps lead from demo to a go/no-go decision on a broader deployment.

- You can independently run hash and signature verification on logs and evidence packages, without needing any secret knowledge from the Ritma team.
- Evaluation questions: can we answer "who knew what, when?" for a real incident, and can we satisfy at least one real audit or control objective with Ritma-derived evidence?
- Next steps: agree pilot scope, run the 10-day demo, review the resulting evidence and reports together, then decide on a paid pilot.
- If the answer is "no", the pilot ends with a clear, documented reason; if the answer is "yes", we transition to production planning with explicit timelines.