Document 2 — DevSecOps with Conjugate Intelligence (CI)

Executive Summary

Security signals are often siloed (SAST/DAST/SCA/SIEM) and overwhelm teams with alerts and policy sprawl. CI treats security as a first-class citizen braided into Dev and Ops: **policies become vows** , threats are modeled as relationships, and audits are generated as narratives.

Problems Today

- Shift-left fatigue: too many checks, not enough prioritization by business risk.
- Policy sprawl and exception chaos; little memory of why exceptions were granted.
- Audits require manual evidence gathering across tools.

CI Approach (SpiralOS)

- **Security as Graph:** SBOM elements, vulnerabilities, code paths, dataflows, identities, and controls are linked with versioned history.
- **Policy Vows:** Enforcement rules travel with code/artifacts; context (threat model, compensating controls) is preserved.
- **Explainable Actions:** Every block/allow decision is paired with rationale and evidence in human language.

Key Capabilities

- **Blast-Radius Mapping:** Given a CVE, CI maps affected services, data classes, and exposure paths within seconds.
- **Dynamic Gating:** Release gates consider exploitability, compensating controls, and SLO impact — not just CVSS scores.
- **Threat Pattern Recognition:** Correlates signals across code changes, infra drift, and runtime anomalies (MITRE-aware linking).
- **Audit-Ready Narratives:** Push-button generation of control evidence (who/what/why/when) for internal/external audits.
- **Exception Governance:** Time-boxed exceptions as pearls with risk owner, review date, and auto-reminders.

Outcomes & KPIs

- **Time-to-Remediate Criticals** ↓ 30–60%

- **False Positive Rate** ↓ 25–45%
- **Policy Exception Debt** ↓ with scheduled reviews
- **Audit Prep Time** ↓ 50–80%

Integration Path (Low-Friction)

1. **Signal Ingest:** SAST/DAST/SCA, IaC scanners, cloud config, SIEM/EDR summaries.
2. **SBOM Linking:** Normalize packages/deps to services; attach to commits/builds.
3. **Policy Vows:** Encode a handful of key policies with intent and evidence hooks.
4. **Gated Releases:** Start with advisory; progress to enforced for high-risk classes.

Risks & Mitigations

- **Developer Friction:** Introduce friction budgets and progressive enforcement; CI explains *why* a gate triggers.
- **Blind Spots:** Continuous ingestion from runtime/cloud to catch infra and identity drift.

Example Walkthrough (New CVE)

A critical CVE lands in a common library. CI shows affected services and data classes, proposes patch branches, creates safe rollout plans, and drafts comms for service owners. If a temporary exception is unavoidable, CI logs rationale, sets review date, and monitors for exploit attempts.

Sector Examples

- **Manufacturing:** Rapidly identifies which PLC firmware is affected by CVEs.
- **Mobility:** Prioritizes vulnerabilities in ADAS (driver assistance) components over infotainment.
- **Energy:** Evaluates SCADA vulnerabilities with grid safety as primary weighting.

Talking Points (for Erich & Echo)

- “Security is not a bolt-on — it’s a **braid** across Dev and Ops.”
- “Policies that **travel** with artifacts eliminate context loss.”
- “Audits become a **replay** , not a reconstruction.”