Meta-Governance

One Platform. Continuous Proof.

Proof. Not Promises.

Modern software systems are no longer static products.
They are living ecosystems of repositories, pipelines, infrastructure, dependencies, data models, AI capabilities, and operational decisions.

Most organizations govern these systems using disconnected dashboards, periodic scans, spreadsheets, and institutional memory.

Meta-Governance was built to replace that fragmentation with continuous, evidence-driven truth.

This is not another security dashboard.

It is a unified governance and evidence platform that continuously discovers, models, verifies, and proves the reality of software-defined systems.

The Meta-Governance Platform

Meta-Governance combines multiple purpose-built systems into a single operational framework:

- Nexus
- GitViz
- Hurricane
- Beacon
- Orpheus
- ERD
- MCPViz
- Argus
- Axiom

Each product answers a different operational question.

Together, they create something far more important:

A continuously reconstructed, cryptographically provable model of software reality.

Nexus

The Discovery Engine

Nexus continuously crawls repositories, infrastructure definitions, CI/CD configurations, dependencies, and non-code artifacts to build a complete digital twin of the repository.

Unlike traditional SBOM tools, Nexus does not stop at package inventories.

It captures:

- Software dependencies
- Infrastructure-as-Code
- Kubernetes and Terraform
- CI/CD workflows
- Documentation and media
- Archives and binaries
- Compliance evidence
- Security findings
- Structural relationships

The result is a graph-based system model grounded directly in source truth.

Nexus produces:

- CycloneDX SBOMs
- ArtifactBOMs
- Merged governance graphs
- Dependency reachability analysis
- Blast-radius intelligence
- SCAP-aligned compliance evidence

This is not scan output.

It is explainable system reconstruction.

GitViz

Governance Through Git Reality

GitViz transforms raw Git history into operational governance intelligence.

Every commit, branch, merge, and file change becomes measurable evidence.

GitViz analyzes:

- Change velocity
- Coupling pressure
- Blast radius
- Churn
- Fragility
- Bus factor risk
- Collaboration topology
- Hidden dependencies
- Stability trends

It provides engineering telemetry derived directly from repository truth—not manually curated metrics.

GitViz allows organizations to answer:

- What changed?
- Who changed it?
- What areas are destabilizing?
- Which teams own critical knowledge?
- Is risk increasing or decreasing over time?

This turns Git into a governance signal, not merely a version-control system.

Hurricane

The Evidence Ledger

Hurricane is the immutable evidence and provenance layer of the platform.

It stores:

- SBOM baselines
- Historical evidence

- Vulnerability correlations
- Compliance findings
- Merkle proofs
- Attestations
- Timeline relationships
- Structural lineage

Hurricane preserves software truth across time.

It enables organizations to replay and reconstruct:

- Historical system state
- Prior vulnerability posture
- Dependency evolution
- Compliance evidence
- Provenance lineage

The result is continuous operational memory instead of isolated snapshots.

Beacon

Frozen Intelligence

Beacon solves one of the hardest problems in software governance:

“What did the world know at the exact time this decision was made?”

Beacon builds cryptographically verifiable Intelligence Packs by merging:

- NVD
- OSV
- GitHub Advisories
- Compliance intelligence

into deterministic, versioned vulnerability datasets.

These packs can operate:

- locally
- air-gapped
- offline
- in classified environments

Beacon eliminates ambiguity caused by constantly shifting vulnerability feeds.

Same SBOM + same Beacon pack = same result. Every time.

That creates reproducible, defensible risk analysis.

Orpheus

The Governance Scholarship Layer

Orpheus captures governance as structured knowledge rather than tribal memory.

Using BPMN-based modeling and metadata-driven governance structures, Orpheus allows organizations to formally model:

- operational governance
- compliance flows
- engineering processes
- decision logic
- lifecycle controls

Orpheus is intentionally separated from runtime execution.

It is not workflow theater.

It is a structured scholarship layer where governance itself becomes versioned, reviewable, and explainable.

ERD

Data Governance and Schema Intelligence

Most governance platforms ignore database evolution entirely.

ERD does not.

The ERD system continuously analyzes:

- schema structure
- migration history
- foreign-key integrity
- dangerous migration patterns
- SQL anti-patterns
- refactor opportunities

It produces deterministic governance evidence for:

- schema baselines
- migration safety
- query risk
- structural integrity

The SQL Advisor identifies:

- cartesian joins
- non-sargable predicates
- unsafe type conversions
- orphan risks
- migration drift

while generating actionable remediation guidance.

This brings governance directly into the database lifecycle.

MCPViz

AI Capability Governance

MCPViz addresses one of the newest governance problems in software:

AI capability visibility.

As AI systems increasingly expose tools, agents, and Model Context Protocol (MCP) capabilities, organizations lack a structured method to inventory and govern those capabilities.

MCPViz solves this by discovering and visualizing:

- MCP servers
- callable capabilities
- tool exposure surfaces

- agent interactions
- capability relationships

This enables the creation of CAP-BOMs (Capability Bills of Materials), extending governance principles beyond software components into AI operational behavior.

Meta-Governance treats AI capability exposure as a governance problem—not merely an AI problem.

Argus

Runtime Reality Verification

Static governance is no longer sufficient.

Argus extends Meta-Governance into runtime verification.

Argus compares:

- approved baselines
- signed manifests
- expected software composition

against:

- actual runtime systems
- live deployment state
- executing processes
- structural drift

This closes the gap between:

“What was approved?”
and
“What is actually running?”

Runtime drift detection transforms governance from a static audit exercise into continuous operational verification.

Axiom

Post-Quantum Proof Infrastructure

Axiom is the cryptographic core of Meta-Governance.

It is a stateless, post-quantum attestation engine built around NIST FIPS 204 ML-DSA-65 cryptography.

Axiom mathematically seals:

- repositories
- deployments
- infrastructure baselines
- firmware payloads
- database records
- compliance artifacts

using post-quantum signatures and deterministic manifests.

Key capabilities include:

- ML-DSA-65 post-quantum signing
- Merkle-tree verification
- air-gapped execution
- BYOK key control
- hardware security integration
- runtime structural attestation

Axiom is designed for:

- defense
- aerospace
- healthcare
- critical infrastructure
- financial systems
- regulated environments

It shifts software assurance from trust to mathematical proof.

The Evidence Loop

Meta-Governance is not a collection of disconnected products.

It is an Evidence Loop.

GitViz captures chronology.

Nexus captures structure.

Beacon captures intelligence.

Hurricane preserves evidence.

Axiom proves integrity.

Argus verifies runtime reality.

Orpheus documents governance intent.

ERD governs data evolution.

MCPViz governs AI capability exposure.

Together, they create:

A living, provable history of software reality.

Not reports.

Not screenshots.

Not compliance theater.

Proof.

Built for High-Assurance Environments

Meta-Governance was engineered specifically for environments where trust assumptions fail.

The platform is:

- local-first
- air-gap capable
- single-binary deployable
- deterministic

- reproducible
- cryptographically verifiable

It supports:

- defense
- aerospace
- federal
- critical infrastructure
- regulated healthcare
- high-security enterprise environments

without requiring SaaS dependency or source-code exfiltration.

The Shift

Most governance systems produce reports.

Meta-Governance produces explainable truth.

Most platforms rely on promises.

Meta-Governance relies on proof.

Most tools answer:

“What do we think is happening?”

Meta-Governance answers:

“What can we mathematically prove?”

Meta-Governance

Truth. Trust. Traceability.

Proof. Not Promises.