Xiaotime *Labs*

Integrated Cyber Risk Governance

Building *Upstream*

Why AI governance starts in the SDLC, not after deployment.

A position paper from Xiaotime Labs on Integrated Cyber Risk Governance (ICRG). The architecture, the argument, and the outcomes that matter.

A position paper.
The architecture, the argument, and the outcomes that matter.

Version 2.0
April 2026

For CISOs, heads of platform, and compliance leaders weighing how to build AI governance infrastructure that scales with product development rather than against it.

The premise

AI governance is becoming a procurement question before it is a regulatory one. Buyers in financial services, healthcare, the public sector, and an expanding set of mid-market verticals now treat AI as a category of risk that requires evidence, not attestation. They are no longer satisfied with a SOC 2 letter from last summer and a slide that says "responsible AI." They want to see the controls operating.

Most teams building AI systems are not prepared for that conversation. Not because they don't care. Because the instrumentation they built for shipping software does not produce the evidence that governance asks for. Code review, scanning, deployment approvals, incident response. These exist in every mature engineering organization. They also emit signals that, if captured, answer almost every compliance question a CISO or auditor can ask. The signals are simply not wired through to the places that need them.

The result is a gap that has been getting wider. Compliance teams chase after engineering with questionnaires. Engineering pushes back because filling in a spreadsheet is not what shipped the feature. An enterprise security review takes months. A deal closes or it doesn't. The system is inefficient for everyone in it.

This is not a process problem. It is an architecture problem.

The upstream thesis

The software development lifecycle already is a governance system. Version control gives you a tamper-evident record of every change. Required reviewers give you authorized approval. Pull-request checks give you policy enforcement. Signed artifacts and software bills of materials give you chain of custody. Identity and access management give you authorization at the seam between code and production.

None of these were designed for AI governance. All of them can be extended to do it. The only thing missing is a control plane that reads the signals the SDLC already produces and projects them into the forms that regulators, auditors, and customers understand.

The thesis is straightforward. AI governance is strongest where AI risk is born. At the point a prompt is written, a model is updated, a dataset is joined, a dependency is added. Governance that starts there produces evidence as a side effect of building software. Governance that starts after deployment produces evidence as a manual project, quarterly, forever.

Most organizations already have the primitives. The work is connective tissue, not a parallel stack.

What an integrated cyber risk fabric looks like

Xiaotime Labs builds a platform called ICRG, Integrated Cyber Risk Governance. It is an implementation of the upstream thesis. We describe it here at the level of capability so that readers building in-house, evaluating vendors, or designing a governance program can take the pattern and apply it regardless of which stack they choose.

1. A single control register

At the center sits one structure. A register that describes every control the organization is responsible for, across every framework that applies. NIST AI RMF, NIST CSF 2.0, ISO/IEC 42001, SOC 2, HITRUST, the AI Control Matrix. All present as projections of the same underlying set of controls, annotated with the mappings between them.

The register records, for each control, how it is validated. By behavior observed in telemetry, by attestation from an owner, or by a hybrid. It records the sources that contribute evidence. It records the current maturity and the last time it moved.

A single register is the design decision that makes everything else possible. When controls are fragmented across tools, you end up mapping the same evidence into five different frameworks by hand. When they are unified, a new framework is a view, not a migration.

2. Telemetry as the source of truth

Evidence is a consequence of observation, not of documentation. If governance depends on someone remembering to update a spreadsheet, it will decay. If it depends on signals the system already emits, it cannot.

The fabric subscribes to the signal sources that matter. The version control system. The CI pipeline. Dependency scanners. Cloud-configuration observers. Identity providers. External attack-surface scanners. And the part specific to AI systems: the operational behavior of the AI models and agents themselves. Prompts. Model versions. Retrieval corpora. Agent tool-call patterns. Output distributions. Guardrail events.

These signals land in an append-only store. They are projected into the control register on arrival. The register is always current. The question "what is our posture right now" has a real answer, not a slide.

3. Policy-aware SDLC

Most of what an AI governance policy needs to do happens inside the development lifecycle. Require peer review on the prompts that touch sensitive data. Block a dependency whose license is incompatible with the product. Flag a model-version change that was not signed off by a risk owner. Require a data-protection impact assessment when a new data source is added.

These are not rules that belong in a GRC tool running a week behind engineering. They belong at the pull request, where the action is actually happening and where the person making the change can still respond. Policy-as-code, run as part of the SDLC, is how governance becomes fast enough to keep up with AI-assisted development.

4. A purpose-built agent layer

The fabric coordinates a small set of specialized agents, not a single general-purpose model, each with a scoped responsibility. One coaches users through requirements articulation. One implements changes against issues and pull requests. One inspects code changes for policy and security considerations. One monitors posture in real time and surfaces emerging risks. One operates the deterministic audit workflow. One acts as the conversational coach for the risk-context interview that establishes an organization's mission, obligations, and impact tolerances.

The point of a multi-agent design is not novelty. It is that governance is a long chain of distinct tasks, each with its own context and authority model. Collapsing the chain into one agent breeds errors.

Scoping each agent to a single kind of work keeps it auditable and keeps human oversight meaningful.

Each agent operates with a scoped persona, a scoped set of tools, and a scoped memory. Nothing in the design resembles a single autonomous system with broad reach. The architecture is deliberately narrow, composable, and inspectable. Governance that is not inspectable is not governance.

Every agent emits structured events. Every decision (allowed, denied, escalated, approved) is captured in the audit trail. The governance of the governance layer is the same governance layer.

5. Continuous correlation

Technical posture is not business posture. A vulnerability in a customer-facing service is not the same as a vulnerability in an internal reporting tool. A dependency with a supply-chain concern affects one team differently than another. The fabric correlates technical signals to the business context captured in the organization's risk profile. Services. Critical assets. Impact tolerances. Regulatory obligations.

Consider a concrete example. A critical vulnerability in a dependency ranks differently depending on where it lives. A library used by a revenue-generating service under PCI scope, serving regulated customers, outranks the same library used by an internal reporting tool that no external party sees. The CVE score is identical. The operational priority is not. The system can show you why, in the language of the business, without anyone having to argue it out.

This is how technical findings become operational priorities without the priority being invented in a meeting. The same correlation drives executive views, board reporting, and the evidence packages that customers and regulators ask for.

What changes when governance is upstream

When the architecture above is in place, a number of things that used to be hard stop being hard.

Audit preparation is a query, not a project. The evidence an auditor needs is continuously produced and always indexed. A question like "show me every privileged-access grant over the

past 90 days, who approved it, and the control it was governed by" is answered in minutes, from the same system that governs the grants.

Security reviews in enterprise sales shorten. The buyer's security team can be handed a live dashboard scoped to their evaluation, with evidence mapped to the frameworks they require. The review stops being a one-time document exchange and starts being a traceable interaction.

Compliance cost scales with the business, not with the compliance program. New frameworks are mappings, not migrations. New controls inherit telemetry from existing sources. The marginal cost of adding SOC 2 to an organization that already has NIST AI RMF coverage is measured in mappings, not in headcount.

AI-specific obligations become tractable. The EU AI Act's high-risk system requirements, lifecycle risk management, technical documentation, record-keeping, transparency, human oversight, all map cleanly onto capabilities the fabric already provides. What reads as a new, heavy regime becomes a projection over an existing evidence base.

Trust becomes a deliverable. Customers do not buy promises. They buy proof. An organization that can present continuously operating controls, not annual attestations, occupies a different position in a sales conversation than one that cannot.

We are intentionally not publishing a table of ROI percentages or sales-cycle multipliers. The claims above are architectural consequences. In practice, the shape of the effect is audit preparation that moves from weeks to hours, and enterprise security reviews that shorten meaningfully rather than marginally. The magnitude of the effect varies by organization. We would rather discuss it against a specific context than against an industry-average number that means nothing to anyone in particular.

Framework alignment

ICRG is designed against the current generation of AI governance frameworks. The table below shows how the fabric's capabilities map onto the major functional requirements in each.

REQUIREMENT	FRAMEWORK	HOW THE FABRIC ADDRESSES IT
Risk management throughout the lifecycle	NIST AI RMF `MAP` / `MANAGE`; EU AI Act Art. 9	Continuous risk correlation from signal through business context. Periodic re-assessment triggered by changes rather than calendars
Technical documentation	EU AI Act Art. 11; ISO/IEC 42001 Clause 8	Documentation derived from system state rather than maintained by hand. Versioned alongside the system it describes
Record-keeping and audit trail	EU AI Act Art. 12; SOC 2 CC8.1; ISO 27001 A.12.1.2	Append-only event record for every policy decision, approval, and state change. Queryable, tamper-evident
Transparency	EU AI Act Art. 13; NIST AI RMF `GOVERN`	Live posture views scoped to the viewer. Evidence packages produced on request
Human oversight	EU AI Act Art. 14; NIST AI RMF `MANAGE`	Policy decisions that cross risk thresholds route to human approvers. Approval events captured with actor and context
Behavior and output monitoring	NIST AI RMF `MEASURE`; AICM MLO / LOG controls	Continuous observation of agent tool calls, model versions, output distributions, guardrail events
Supply-chain transparency	NIST AI RMF `GOVERN-1.4`; ISO 27001 A.15	Dependency and software-bill-of-materials tracking integrated with the control register
Access management	NIST CSF `PR.AC`; SOC 2 CC6	Identity telemetry correlated through the SDLC to production, with drift detection and periodic review triggers

The alignment is not incidental. The fabric was designed against these frameworks from the outset, not retrofitted to them.

A pragmatic note on rollout

Organizations adopting this architecture do not need to do it all at once. The sequence that works is usually this.

First, consolidate the control register. Pull whatever controls you already manage, typically scattered across GRC tools, spreadsheets, and security team tribal knowledge, into a single representation. This is the hardest social step. The technical part is mostly data modeling.

Next, connect the cheapest telemetry. Version control events. CI results. Dependency scanning. Cloud configuration. These sources usually exist. The work is subscribing to them and projecting the signals into the register.

Then extend to AI-specific observation. Prompt provenance. Model version tracking. Agent telemetry. Guardrail events. This is where the AI-aware part of the architecture earns its keep.

Finally, move policy into the SDLC itself. Pre-merge checks. Approval routing. Exception handling. This is where governance stops being something that happens to engineering and becomes something engineering does.

The path is sequential because each step produces value and establishes trust that supports the next. Organizations that try to do everything at once tend to succeed at nothing. Organizations that start with policy enforcement before they have an evidence base tend to produce compliance theater. The order matters.

About Xiaotime Labs

Xiaotime Labs builds ICRG, Integrated Cyber Risk Governance, the platform this paper describes. We work with organizations that are building AI systems and need governance infrastructure that scales with their product rather than against it. Our focus is mid-market and enterprise teams whose AI deployments are real enough to be subject to scrutiny and early enough that the right architecture can still be chosen.

Xiaotime Labs was built by Craig George and Stephan Hundley, active practitioners who have known each other for nearly fifteen years. One a working account executive. The other a veteran CISO engineer. They built ICRG because they needed it.

We are happy to talk about specifics under NDA. What we have built, where it is deployed, and how it would apply to your environment. This document is the argument. The conversation is where the fit gets assessed.

Next steps

If your AI systems need to operate in regulated environments, or if your buyers are asking for governance evidence you do not yet have, we would like to talk.

Contact · info@xiaotimelabs.ai
Learn more · xiaotimelabs.ai/icrg.html

A technical companion to this paper, covering architecture and implementation detail, is available to prospective customers under NDA.