

POSITION PAPER • THE GOVERNANCE PRIMITIVE

Estate as Code.

Capturing the whole estate, including the AI, as code you can actually govern.

You cannot govern what you have never fully seen. Most security programs run on an inventory, and an inventory is a photograph: accurate the moment it is taken, stale the moment anything changes. Estate as Code captures the running estate as versioned, read-only code, and keeps it reconciled with reality. This paper is the argument for why that is the primitive continuous governance has been missing.

PAPER

FROM

AUDIENCE

Position paper

Xiaotime Labs

CISO · Security leaders

Estate as Code is the practice of capturing your entire running estate, every cloud account, identity, data store, endpoint, and AI workload, as versioned read-only code, and keeping it reconciled with reality. Most security leaders have not heard the term yet. Here is why it is about to matter.

You cannot govern what you cannot see

Every security program rests on a quiet assumption: that someone, somewhere, has an accurate picture of what the organization actually runs. That picture is almost never accurate. The cloud account count grew faster than the wiki. The SaaS list lives in three spreadsheets and a procurement inbox. The endpoint inventory is a quarter behind reality. And a new plane sits on top of all of it now, the AI plane, where models, assistants, and autonomous agents reach into real systems on someone's behalf, often without a line item in any register at all.

The industry's answer has been the inventory: the configuration database, the asset spreadsheet, the annual scoping exercise. An inventory is a photograph. It is accurate the moment it is taken and stale the moment anything changes, which is constantly. Governance built on a photograph is governance built on a guess.

Estate as Code starts from a different premise. The estate is not a list you maintain by hand. It is a living thing you capture as code, read-only, and then keep reconciled against reality the same way you keep code reconciled: with diffs.

The estate has outgrown the inventory

For most of the cloud era, "the estate" meant infrastructure: the accounts, the networks, the workloads. That definition is now incomplete, and the incomplete part is the part with the least governance and the fastest growth. A modern estate spans four planes.

- ◆ **Platform.** The cloud infrastructure and managed services. The part everyone already calls the estate.
- ◆ **Endpoint.** The devices and their management posture. The part that walks out the door.
- ◆ **Provider.** The AI model platforms. The hosted models, the inference endpoints, the knowledge bases an assistant can read.
- ◆ **Configuration.** The manifests that define behavior. The agent personas, the policies, the workflow definitions that decide what software and AI are allowed to do.

An agent does not live on one plane. It runs on the platform, it may reach an endpoint, it calls a provider's model, and its behavior is set by configuration. If your capture only covers infrastructure, three of the four planes an AI agent touches are invisible to your governance. Estate as Code captures all four, as code, in one description you can reason about together.

Why as code, and why now

There is a quiet movement in software engineering called Everything as Code. The idea is plain: anything you would otherwise manage by hand becomes a file you can version, review, and test. It started with infrastructure and kept going, into configuration, networks, pipelines, and documentation. It works where it reaches, because code is the one medium that is reviewable, diffable, and auditable by default.

Two things about that movement matter here. It is younger than it looks, and most teams are early in it. And where it has been applied, it has mostly been applied to the new things an organization builds, not to the estate it already runs. The greenfield is code. The inherited estate, the one with the actual risk in it, is still prose: a questionnaire, a screen-share, and a security engineer transcribing what they see into a spreadsheet.

Estate as Code is that idea turned on the estate itself. It captures what is already running, without anyone rebuilding it, as versioned and diffable code across all four planes. Not a model of what was designed. A capture of what exists. The velocity case for the broader movement is the subject of our companion paper Everything as Code; the case that governance belongs in the pipeline rather than the report is GRCDevSecOps. This paper is about the part of the estate both of them assume you can already see.

What as code actually buys the CISO

Calling the estate "code" is not a metaphor. It is a set of properties a photograph does not have.

- ◆ **Diffable.** A capture taken today and a capture taken next week produce a diff. Drift stops being something you discover during an audit and becomes something you see the moment it happens. A new public data store, a loosened identity policy, a model endpoint that appeared over the weekend: each is a line in a diff, not a surprise in a quarterly review.
- ◆ **Reviewable.** Code is the medium security teams already know how to inspect. The estate expressed as code can be read, questioned, and reasoned about by a person or by an agent, using the same review discipline the rest of the as-code stack already uses.
- ◆ **Reconcilable.** A capture is not a one-time export. It is continuously checked against the live estate. The description and the reality are held together on purpose, so the description never quietly rots into another stale inventory.
- ◆ **Projectable.** This is the part that turns capture into governance. Once the estate is code, every element can be projected through the control frameworks that apply: NIST CSF, ISO 27001, SOC 2, HITRUST, and the AI-specific frameworks that now matter, the NIST AI Risk Management Framework and the Cloud

Security Alliance AI Controls Matrix. A control stops being a question you ask a human once a year. It becomes an assertion checked against the captured estate continuously. Projecting every observation through every framework at once, and advancing findings through a closed remediation pipeline, is the runtime we describe in Closing the Loop. Estate as Code is what that runtime sees.

Read-only and keyless, by design

A CISO's first question about anything that touches the whole estate is the right one: what can it change, and what credentials does it hold.

Estate as Code is read-only. The capture observes; it does not modify. It cannot reconfigure a network, open a port, or alter an identity, because it is not granted the permission to. The point of the exercise is to see the estate truthfully, and a tool that can change what it is measuring is a tool you cannot trust to measure.

It is also built to avoid standing credentials. Access is scoped to the minimum needed to observe, and granted through short-lived, provider-issued identity rather than long-lived keys sitting in a configuration file. The estate capture is one of the most sensitive integrations a platform can ask for, precisely because it sees everything. The right posture for that integration is least privilege, read-only, and no secret to steal.

What changes when the estate is code

Onboarding compresses from weeks to a day. The discovery that used to be a manual transcription becomes a capture. The first conversation with a new environment starts from an accurate, current picture instead of a questionnaire.

Coverage becomes provable. "Are we governing everything" stops being a matter of faith. The captured estate is the denominator. What is governed, and what is not yet, is a number you can show, not a feeling you defend.

Drift is caught as it happens. Because reconciliation is continuous and drift is a diff, the gap between a change and its detection collapses. The estate does not drift for a quarter before anyone notices.

The AI plane joins the same governance as everything else. A fleet of agents and a set of model endpoints are a new attack surface and a new audit surface at once. Captured as code on the same planes as the infrastructure, they are governed by the same controls, reviewed the same way, and accounted for in the same record. The AI is not a separate program bolted on beside the security program. It is in the estate, so it is in scope. Why a fleet of agents needs one mediated control point, so that attribution and policy become properties of every call, is the argument of our paper The Governed Fleet. Estate as Code is how those agents and their model endpoints enter the estate in the first place.

THE SHIFT

Evidence stops being something humans collect **against a deadline.**

When the estate is code and controls are assertions against it, the capture, the diffs, and the control results are the evidence. Governing the estate and evidencing the estate become the same act.

The objections, answered plainly

"This is just cloud posture management, or a configuration database with better marketing."

No. A posture tool scores a cloud against a checklist and shows you a dashboard. A configuration database stores a list a human curates. Estate as Code captures the estate as versioned, diffable code across all four planes, including the AI, and projects it through every framework that applies. The output is not a score on a screen. It is a description you can review, reconcile, and govern, and that an agent can act on. The dashboard is the smallest part of the value.

"Read-only, really? Everything wants write access eventually."

Read-only is the design, not a phase. Capture and governance are observation problems. The execution that does change things, the remediation that ships a fix, lives in a separate pipeline with its own review gates and a human authorization at the one place it matters. Seeing and changing are deliberately kept apart.

"Our estate is too large and too dynamic to capture."

Large and dynamic is the argument for capturing it as code, not against. A spreadsheet cannot keep up with a dynamic estate. A continuously reconciled, diffable capture is built for exactly that. The harder the estate is to hold in a human's head, the more it needs to be held as code.

"We already have an inventory."

You have a photograph. The question is not whether you have a list. It is whether the list is true right now, whether it includes the AI plane, and whether your controls run against it automatically. An inventory answers none of those. The estate as code answers all three.

Where Estate as Code sits

Estate as Code is the discovery anchor of the Cyber Risk Fabric. It is how the fabric sees. The capture feeds the control register, where the estate is projected through the frameworks that apply and governance becomes continuous rather than periodic. When something must change, the execution face of the same pipeline carries the fix through review to a single human authorization.

This is the through-line of everything Xiaotime Labs builds. Governance is not a report produced after the work. It is woven into how the estate is captured, how it is reasoned about, and how it is changed. Capture the estate as code, including the AI, and governance stops being the thing you scramble to assemble before an audit. It becomes a property of the estate itself.

You cannot govern what you cannot see. Estate as Code is how you see all of it, and keep seeing it, so that governing it is finally something you do continuously instead of once a year.

Related reading

- ◆ **GRCDevSecOps.** Why governance belongs in the pipeline, not the report.
- ◆ **Building Upstream.** Why AI governance starts in the SDLC, not after deployment.
- ◆ **Closing the Loop.** The continuous monitoring runtime that Estate as Code feeds.
- ◆ **Everything as Code.** The velocity case for moving governance into the code layer.
- ◆ **The Governed Fleet.** Why a fleet of AI agents is only as trustworthy as the control point they share.
- ◆ **DoctorWhiskers Explains Everything as Code.** How AI agents author, operate, and govern the as-code stack.