Xiaotime Labs

POSITION PAPER · THE CATEGORY

ISMS as Code.

The security program, running as code. Not described. Operated.

Most tools help you describe your ISMS. We are your ISMS.

PAPER	FROM	AUDIENCE
Position paper	**Xiaotime Labs**	**CISO · Security leaders · Board**

Every security leader has an information security management system. Very few can show you one running. Ask to see it and you get a folder: a scope statement, a risk methodology, a policy set, a statement of applicability, last year's internal audit report. The folder is accurate the day it is approved. The program it describes changes every day after that.

ISMS as Code runs the whole management system as code: declared, versioned, continuously enforced and continuously evidenced. Xiaotime built it, and we run our own security program on it. This paper explains why the ISMS is the next thing Everything as Code reaches, what that gives a security program, and what it lets you stop paying for.

The ISMS was always meant to run

Start with the definition, because the standard is clearer than the market.

An ISMS is not a document. ISO/IEC 27001 asks an organization to *establish, implement, maintain and continually improve* it (clause 4.4). Every verb in that sentence describes something running. Establishing it means deciding scope and risk appetite. Implementing it means operating controls. Maintaining it means keeping those controls true as the organization changes. Improving it means measuring, finding what failed and fixing it.

That is a loop. The standard describes a system with a heartbeat.

The industry has known the loop should be continuous for a long time. NIST wrote down continuous monitoring as an organization-wide program in 2011 (SP 800-137). HITRUST, SOC 2 and the newer AI frameworks describe the same destination in their own words. What most organizations operate today is something else: an annual cycle, a sample, a binder that gets refreshed before the auditor arrives.

The gap between the definition and the practice is not negligence. It is what happens when a running system has to be operated by hand.

Everything as Code is the operating model

Engineering solved this problem once already.

Infrastructure used to be hand-built. Then it became code: declared in files, versioned, reviewed, applied by a pipeline. Configuration followed. Networks followed. Policy is following. Each time, the same thing happened. The artifact moved into version control, the pipeline took over the change, and the speed at which the organization could change it, safely, went from weeks to minutes.

The idea has a name: Everything as Code. It is an operating model, not a tool. Anything you would otherwise manage by hand becomes something you can declare, review, test and reconcile against reality.

Governance has been the slowest to move. Our companion paper, *Everything as Code, including governance*, makes the velocity case. Most teams have not even reached policy as code. Their policies still live in PDFs and their proof is an annual sample.

ISMS as Code takes the operating model one step further than governance. It does not put a policy or a control into code. It puts *the management system itself* into code: the scope, the risk method, the control set, the monitoring, the evidence, the review. The whole program, running.

The market enables it. Nobody is it.

The tooling around the ISMS is good, and it is growing. Each category does real work.

Compliance automation platforms (Vanta and Drata among them) connect to an organization's systems, collect evidence against frameworks and keep track of what is due.

Document-first ISMS platforms (ISMS.online among them) give the program a structured home: the scope, the policies, the risk register and the statement of applicability, organized the way an ISO auditor expects to read them.

Cloud-native application protection platforms, the CNAPPs (Wiz and its peers), secure the cloud layer: configuration, workloads, identities, exposure.

Alongside them sit the endpoint platforms, the SIEM, the identity provider and the code scanners. Most security programs run several of them, and run them well.

Look at what each one does. It helps the organization *describe* the program, *prove* part of it, or *secure* a layer underneath it. None of them is the program. The management system, the thing that decides what matters, binds each control to a risk and an obligation, watches whether the control is true, and changes course when it is not, still lives between the tools, held together by people and spreadsheets.

THE CATEGORY

They enable it. Nobody is it.

That is the category Xiaotime built. Not compliance as code, which automates the evidence for a framework. Not GRC automation, which speeds up the workflow around the program. ISMS as Code is the program itself.

A category distinction, not a competitive one

This matters most for the cloud layer, so it is worth saying plainly. A CNAPP secures cloud-native infrastructure. An ISMS governs the security of the whole organization. They are not alternatives. The CNAPP is one of the most valuable signals the ISMS reads.

The same is true of every tool in the stack. In ISMS as Code, the tools an organization already runs sit *under* the management system, not beside it. Each signal they produce is mapped to a control. Each control is mapped to a risk, an obligation and an audit requirement. The tools keep doing what they do best. The ISMS finally has a place to run.

What ISMS as Code means

ISMS as Code is the information security management system declared as code, continuously enforced and continuously evidenced.

Three properties define it.

Declared. The scope, the risk method, the control set, the policies and the people accountable for them are versioned artifacts. A change to the program is a reviewed change, with an author, a reason and a diff, the same way a change to infrastructure is.

Continuously enforced. Controls are not described as operating. They are checked against the systems they govern, all the time, and the change pipeline enforces the policies at the point where work ships. A control that stops being true is visible the moment it stops, not at the next sample.

Continuously evidenced. The evidence is a byproduct of the program running, not a project before the audit. It is generated by the systems that run the controls, time-stamped, append-only and signed, so an auditor can query it rather than request it.

Put those together and the ISMS stops being something you assemble for an assessment. It is something you operate, and something you can show.

Why you want it, and why you need it

Why you want it

One program instead of a dozen projects. Today every framework, every assessment and every customer security review is its own effort, and each one reassembles the same facts by hand. Run the program once, as code, and every one of them is answered from the same record. The fifth framework is a mapping, not a fifth program.

Spending you can tie to risk. Every signal from every tool lands on a control, and every control on a risk and an obligation. For the first time you can see what each tool you pay for actually contributes to the program, and what nothing covers at all.

A team that runs the program instead of documenting it. The hours that go into collecting screenshots, chasing owners and rebuilding the same evidence every year go back into designing controls, reducing risk and improving the program. The work moves up the value chain.

An answer you can give today. When the board, a regulator or a buyer asks whether a control is operating, the answer comes from the running system, not from the last assessment.

Why you need it

The estate moves faster than the cycle. Cloud configuration changes daily. Identities are created and forgotten. AI agents now act on real systems on someone's behalf. A program verified once a year is unverified for most of the year, and the gaps open in exactly the months nobody is looking.

Manual governance has no leverage. Each new framework, regulation and customer requirement adds the same amount of work as the last one, so the cost of the program grows with every obligation. Headcount or delay scales linearly. A program that runs as code is the only version where the next obligation costs less than the previous one.

Proof is becoming the requirement. Buyers, regulators and boards are moving from asking whether a policy exists to asking to see the control operating. A management system kept in documents can describe a control. Only a running one can show it.

Faster, better, stronger

Faster. Changes stop waiting on a compliance review, because the policy runs as a gate in the pipeline the change already goes through. An assessment or a buyer's security questionnaire is answered from the record the program keeps, not from a month of collection. Taking on a new framework is a mapping exercise measured in days, not a program measured in quarters.

Better. Decisions are made on current evidence instead of last year's sample. Every control has an owner, a test and a result, so nothing falls quietly into the gap between two tools. The same gate applies whether a person, an AI agent, a contractor or a dependency bot made the change, so quality does not depend on who shipped it.

Stronger. Drift is caught when it happens, not at the next assessment. Every change to systems and to the program itself is governed and recorded in a signed, append-only record an auditor can trust. And the team's capacity grows without its headcount: the platform's agents do the repetitive work, collecting, mapping, first-pass review, while your people keep the decisions that need judgment.

What it replaces, over time

ISMS as Code does not ask you to rip anything out. It does change what you need to keep paying for.

The tools that protect and detect stay. The CNAPP, the endpoint platform, the SIEM and the identity provider are the signals the ISMS runs on. They do their jobs better when every finding they raise is tied to a control, a risk and an owner.

The tools that describe or evidence the program consolidate. Document-first ISMS platforms, compliance automation, GRC workflow systems, the risk and control spreadsheets, and the evidence-collection projects that surround an audit all exist to write the program down or prove it after the fact. When the program itself is the running, signed record, a separate system to describe it is a second copy of the same truth. Those tools can be retired as their renewals come up.

The overlap becomes visible. Because every control knows which tools prove it, the register shows where two products cover the same control and where a product covers nothing the program needs. Renewal decisions stop being a guess. You stop paying twice for a control you already cover.

The path is gradual by design. Run it alongside what you have. Let the register prove the coverage. Retire what it makes redundant, one renewal at a time.

How it runs

ISMS as Code is not one new tool added to the stack. It is the layer above the stack. Xiaotime's platform is that layer, running in production today, and our own security program runs on it. It has four parts.

The estate, captured as code

You cannot govern what you have never fully seen. The ISMS starts from an accurate, reconciled picture of what the organization runs: cloud accounts, identities, endpoints, data, and the AI workloads that now reach into all of them. Captured as code, the estate produces a diff when it changes. Scope stops being a paragraph and becomes a query. (Our paper *Estate as Code* covers this primitive.)

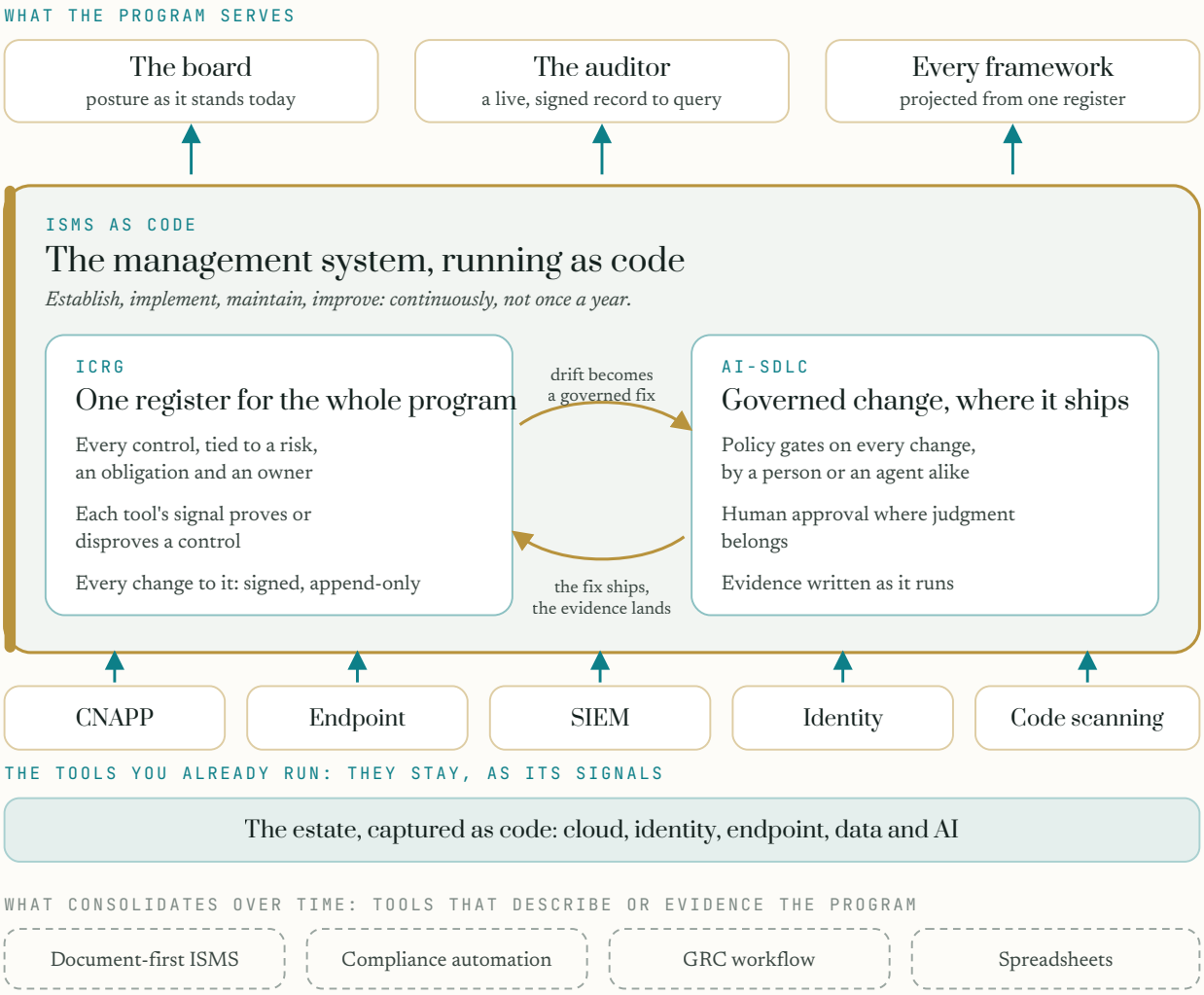
ICRG: one register for the whole program

Integrated Cyber Risk Governance is the management system's core. A single register holds every control the organization is bound to and projects it to every framework at once, so the Nth framework does not cost what the first one did. It reads telemetry from the tools already deployed, the CNAPP, the endpoint platform, the SIEM, the identity provider, and maps each observation to the controls it proves or disproves. Risk is tracked against the same register, so a failing control is also a risk with an owner. Everything that changes the register lands in a signed, append-only audit record.

AI-SDLC: governed change, at the point of change

There is no ISMS without governance, risk and compliance, and an ISMS cannot run as code while its GRC runs weeks behind engineering. So ISMS as Code runs on GRCDevSecOps: GRC working inside the same pipeline as development, security and operations. (Our paper *GRCDevSecOps* makes that case.)

An ISMS that only observes is half a loop. The *AI-SDLC* pipeline is where the program acts. Every change to software, infrastructure or policy passes through the same gates: policy checks, automated review, human approval where it belongs. It does not matter whether a person, an AI agent, a contractor or a dependency bot opened the change. The pipeline writes its own evidence as it runs, so governed change and proof of governed change are one motion.



The ISMS running as code: the tools you run stay as its signals, one register and one governed pipeline run the program, and the board, the auditor and every framework read from it.

Continuous control monitoring, across every tool

The two runtimes close the loop the standard describes. Telemetry flows up from every tool into the register. Drift becomes a finding. A finding becomes a governed change. The change ships through the pipeline and the evidence lands in the record. The register updates, and the posture the board sees is the posture the systems are actually in.

Establish, implement, maintain, improve. Running continuously, as code.

What changes for the CISO

The program becomes something you can point at. Not the binder, the running system.

Audit readiness is a property, not an event. The evidence already exists, generated as the program runs. An assessment is a review of a live record, not a collection project.

The Nth framework is cheap. One register projects to every framework the organization is bound by. Adding one is a mapping exercise, not a new program with new headcount.

The tools you bought finally compound. Every signal from the stack you already run is tied to a control, a risk and an obligation. The CNAPP's finding is no longer just a cloud ticket. It is a fact about the management system.

Change is governed where it happens. The program's policies are enforced at the pipeline, not reviewed after the fact. Human authority stays exactly where it should: at the decisions that need judgment.

What changes for the board

The board asks a simple question: are we in control of our security risk? Today the honest answer is usually "as of our last assessment."

With ISMS as Code, the answer is current. The board view of posture reads from the same record the auditor queries and the security team operates. The organization stops reporting last quarter's snapshot. It reports the program as it stands.

Objections we hear

"We already have an ISMS."

You almost certainly do. The question is where it runs. If it lives in documents refreshed for an assessment, it describes a program that changes daily. ISMS as Code keeps what the organization has decided, its scope, its risk appetite, its policies, and makes the program run against the systems it governs.

"We already bought compliance automation."

Keep it while it serves you. Evidence collection for a framework is one part of the program. ISMS as Code is the program: the scope, the risk method, the controls across every framework, the governed change and the review that improves it. Once the running record covers the evidence, the second copy becomes a renewal decision you can make on data.

"Isn't this what our CNAPP does?"

A CNAPP secures the cloud layer, and does it well. The ISMS governs the whole organization's security, including the cloud layer. The CNAPP becomes one of the most important inputs to it. Different category, same side.

"This sounds like a large transformation."

It is connection, not construction. The register reads from tools already deployed. The pipeline starts against one repository. The program's existing decisions become its first declarations. Coverage grows from there.

"This sounds like a vendor pitch."

It is one. We built ISMS as Code because we needed it, and our own security program runs on it every day: the same register, the same governed pipeline and the same signed record this paper describes. Hold us to it.

The ISMS, running

The information security management system was always meant to run. The standard says so in its first verbs: establish, implement, maintain, improve. The industry built excellent tools around it and left the system itself to people and folders.

Everything as Code gave engineering a way to make the things it manages real, current and reviewable. ISMS as Code applies the same operating model to the security program as a whole. Xiaotime built it, and it is running.

Most tools help you describe your ISMS. We are your ISMS.

Bring us your environment and we will show you your ISMS running.

Related reading

- **Everything as Code.** The velocity case for moving governance into the code layer.
- **Estate as Code.** Capturing the whole estate, including the AI, as code you can govern.
- **GRCDevSecOps.** Why governance belongs in the pipeline, not the report.
- **Closing the Loop.** The continuous monitoring runtime the industry described and nobody built.

Sources. ISO/IEC 27001:2022, clause 4.4 (Information security management system). NIST Special Publication 800-137, *Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations*, September 2011.