Island

AI Playbook for Security Teams

Your guide to evaluating AI risk, closing enterprise security gaps, and confidently saying “yes” to AI at scale.

Agents are joining your workforce, working alongside your people at a breakneck pace. Security teams need a new playbook so they can say “yes” to AI without losing control.

THE CHALLENGE FOR SECURITY TEAMS

Your workforce just got a lot bigger. And a lot harder to govern.

Your organization has a sanctioned AI provider. Maybe two or three. But that only covers a sliver of what's actually happening. Agents are being built, bought, and deployed across every function, and most of them arrived without going through any kind of onboarding. They don't know your data policies, your compliance requirements, or the unwritten rules your employees picked up on day one. They're goal-oriented and relentless, and they'll find a way to complete their mission.

Meanwhile, employees are signing into personal AI accounts, installing AI extensions, and building their own apps with vibe-coding tools. Executives are pushing AI into every function without a unified framework. The sprawl is unprecedented, and it's no longer just about tools — it's about a second workforce operating inside your environment with no registry, no policy, and no audit trail.

Legacy DLP, endpoint protection, and network monitoring were built for a world where applications didn't read content, observe behavior, retain context, connect to internal systems, or take autonomous action. Agents do all of these things, across every surface, at scale.

Three gaps security teams are dealing with right now

Visibility gaps

No single view into which agents, AI tools, models, or tenants are operating across your environment, or what they're actually doing.

Control gaps

Agents have no identity, no scoped permissions, and no audit trail by default. Getting to full coverage requires patching together 7 or more tools that weren't built for an agentic world.

Data exposure

Agents paste, prompt, call tools, and move data across systems in ways traditional DLP wasn't designed to catch. Corporate data moves to ungoverned environments. Model training on sensitive inputs goes unchecked.

This playbook gives security teams a practical framework for governing a workforce that's both human and agentic: understanding where the gaps are, what real control looks like, and how to say "yes" to AI without losing visibility over what it does.

WHAT'S INSIDE

Your roadmap to enterprise AI security

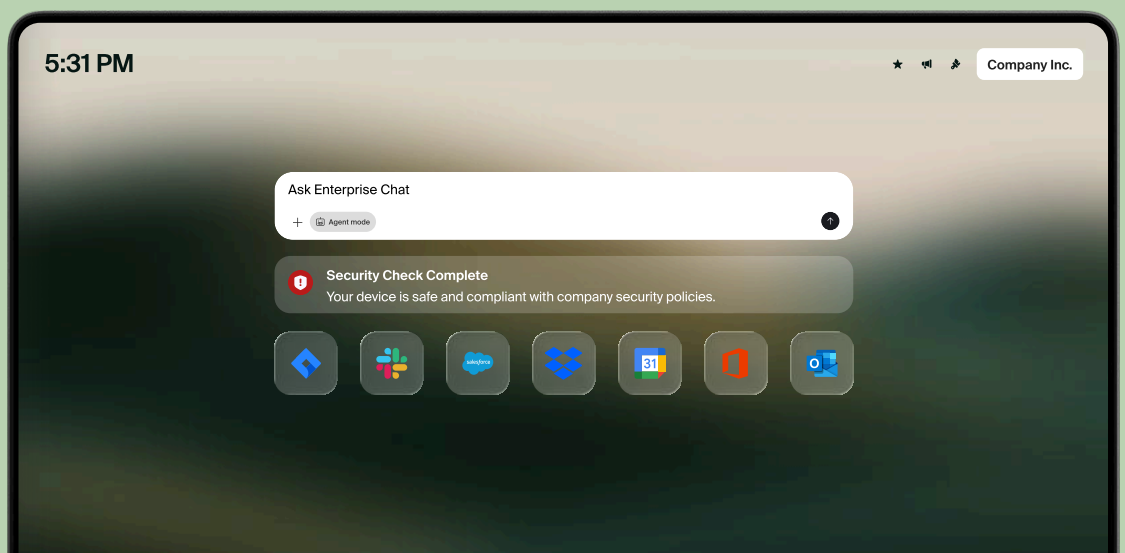
1

The New AI Threat Surface

AI is not arriving through a single door, and neither are agents. Both are entering your environment through nine simultaneous entry points, most of which weren't on your radar when you last updated your security architecture. The threat surface didn't just expand — it changed shape.

Where AI enters the enterprise today

- **Browser-based AI (web destinations)** Consumer AI applications accessed through any browser, including personal accounts for tools like Claude, ChatGPT, and Gemini. Most organizations have no way to distinguish corporate tenant usage from personal usage.
- **Browser extensions** AI-powered extensions that can read page content, inject into prompts, and exfiltrate data. Risk profiles change dynamically as extension behavior evolves.
- **Agentic AI browsers** Emerging browsers like Atlas and Comet ship with AI natively embedded and operate outside corporate policy by default.
- **AI agents** Autonomous software that reads content, takes action across systems, and retains context across sessions. A single employee may now oversee thousands of them simultaneously, each one reaching across browser, desktop, and internal systems to complete its mission.
- **Citizen-built and vibe-coded apps** Employees are building and publishing apps with tools like Claude Code, Lovable, and Cursor faster than any review process can absorb. Most are deployed to public infrastructure with no SSO, no logging, and no data protection.
- **Desktop AI applications** Tools like the ChatGPT desktop app and Claude Cowork operate at the OS level, outside browser visibility and standard DLP coverage.
- **AI-powered IDEs** Development environments with AI built in, like Cursor and GitHub Copilot, operate at the code level with access to sensitive repositories, internal tooling, and proprietary logic.
- **AI connections to enterprise systems** MCP integrations and AI connectors that move data between AI tools and internal applications like Slack, Salesforce, and internal databases.
- **Network-level AI traffic** AI activity traversing the network that point security tools may not capture or attribute to specific users, tools, or tenants. This includes AI coding tools like Claude Code, and productivity tools like Claude Cowork, that operate outside the browser entirely.



Why legacy controls fall short

Legacy security tools were designed to protect a perimeter, but AI collapses the perimeter. Traditional DLP can't see what employees are typing into AI prompts. Endpoint tools don't understand model training implications. Network monitoring can't distinguish between corporate and personal AI tenant usage. And no single legacy tool covers all nine entry points listed above.

The result is a forced choice most security leaders are already confronting: block AI usage and slow down the business, or accept risk and hope the gaps don't get exploited. Neither is a real strategy.

Why other tools can't close the gap

Every vendor approached this problem through the lens of what they already had:

- Network tools see the wire, not what's happening on the endpoint.
- EDR sees the endpoint, not the prompt or the intent behind it, and nothing inside a browser.
- Identity tools see which credential touched a resource, not why.
- CASB and SaaS security tools see the service-side action, not the context behind it.

Each of these sees something real, but none sees the full chain: from the prompt that triggered an action, through the tool calls and responses, to the data that moved as a result. That gap is why incidents get discovered weeks after the fact, by someone piecing together logs from four different tools, instead of at the moment something went wrong.

In an agentic world, that delay isn't acceptable. Agents don't wait. They act, they compound, and by the time a fragmented security stack catches up, the blast radius is already set.

The Risks of Ungoverned AI

Blocking AI isn't the answer. But neither is letting it run without controls. Most organizations sit somewhere in between, and that middle ground is where the risk lives, because AI is running, but controls are incomplete. Here's what ungoverned AI actually looks like in practice.

What ungoverned AI actually looks like for an organization:

Agent actions without guardrails

Agents that operate without scoped permissions and human oversight can leak sensitive data, execute destructive actions, or reach resources they were never meant to touch. Unlike a human making a mistake, an agent can make the same mistake thousands of times before anyone notices.

Excessive token usage and runaway costs

Without visibility into which agents are running, which models they're using, and how often, AI spend scales faster than anyone can track. Token costs compound quickly across teams and projects, and by the time the invoice arrives, the damage is done.

Overly permissive agent access

Most agents are provisioned with far broader access than they need to complete their task. That excess becomes a liability the moment something goes wrong, whether through a prompt injection attack, a compromised MCP server, or an agent simply doing exactly what it was told in ways no one anticipated.

Malicious and vulnerable MCP servers

MCP integrations connect agents to the tools and data they need to work. They also introduce a new attack surface. A malicious or poorly secured MCP server can intercept data, manipulate agent behavior, or serve as a vector for prompt injection at scale.

Shadow AI and ungoverned personal use

Employees using personal AI accounts, unvetted extensions, and consumer-grade tools are moving corporate data into environments the organization has never seen and cannot govern. What looks like a productivity win is often a data exposure waiting to surface.

The common thread across all of these is visibility. You can't govern what you can't see. And in an agentic world, what you can't see is moving faster than ever.

3

Nine Entry Points Security Teams Must Cover

A complete AI security strategy requires coverage across every entry point where AI and agents enter or operate in your environment. Most organizations have partial coverage at best, and almost none have thought through what agent governance truly requires at each entry point. Each of these entry points carries its own version of the risks outlined in the previous section. The questions below will help you identify where your coverage is weakest.

9 AI entry points security teams must cover

1 Browser-Based AI

- Can you distinguish corporate AI tenant usage from personal accounts?
- Do you have visibility into what data employees are sending to each tool?
- Can you enforce policy on prompt content before it leaves the browser?

2 AI Browser Extensions

- Do you have a real-time inventory of every AI extension installed across the organization?
- Can you assess extension risk dynamically as extension behavior changes?
- Can you block or modify extensions based on risk profile?

3 Emerging AI Browsers

- Does your security posture extend to employees who switch to AI-native browsers like Atlas or Comet?
- Can you maintain visibility and policy enforcement outside your managed browser environment?

4 AI Agents

- Do your AI agents have assigned identities with defined permissions?
- Is every agent action fully audited?
- Do you have the ability to apply human-in-the-loop controls for sensitive operations?
- Do you have a handler or orchestrator model in place, with a named human accountable for agents running at scale?
- Is agent cost and token usage tracked per user, team, and project so spend is visible before it hits the invoice?

5 Citizen-Built and Vibe-Coded Apps

- Do you know how many employee-built apps exist in your environment and where they're running?
- Are those apps being published to public or internal infrastructure?
- Can you govern vibe coding apps like Lovable, Base44, Claude Code, and Replit that generate and deploy code autonomously on behalf of users?
- Do they automatically inherit SSO, DLP, and data protection when deployed?
- Is your current review process capable of absorbing the volume of apps employees can now build?

6 Desktop AI Apps

- Does your endpoint coverage extend to the data flowing through tools like the ChatGPT desktop app?
- Can you apply the same DLP policies that govern browser-based AI to desktop AI applications?

7 AI-Powered IDEs

- Do you have visibility into what code and internal tooling AI-powered IDEs like Cursor and GitHub Copilot can access?
- Can you govern what sensitive repositories and proprietary logic these tools are exposed to?

8 MCP Integrations and AI Connectors

- Can you see MCP tool calls that move data between applications like Slack and ChatGPT?
- Can you govern which MCP integrations agents are permitted to use?
- Do you have visibility into what company knowledge users are accessing via AI connectors?

9 Network-Level AI Traffic

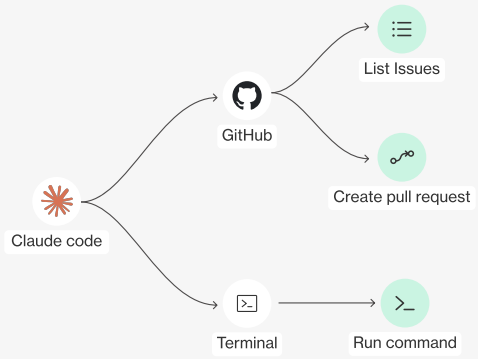
- Is AI activity traversing your network captured & attributed to specific users, tools, & tenants?
- Does your coverage extend to tools that operate entirely outside the browser, like Claude Code and Claude Cowork?
- Can you maintain visibility across roaming users and off-network devices?

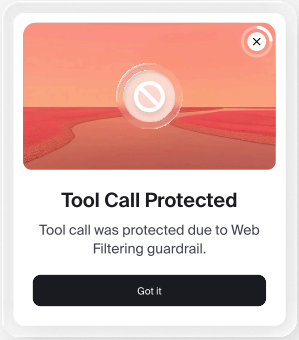
What Enterprise AI Controls *Actually Look Like*

When enterprises have full AI coverage in place, it looks nothing like patching together point solutions. Here is what mature AI security controls deliver across six key dimensions, including what it actually takes to govern agents and the apps your employees are already building.

Agent Visibility and Inventory

- Full inventory of every agent running in your environment, regardless of vendor, including installed MCP servers, skills, code packages, browser extensions, and IDE extensions
- Continuous monitoring of agent behavior so you know not just what agents exist, but what they're actively doing
- Detection of unauthorized or unvetted agents before they reach internal systems or sensitive data
- Real-time visibility into multi-step agentic tasks as a single mission, not a series of disconnected events



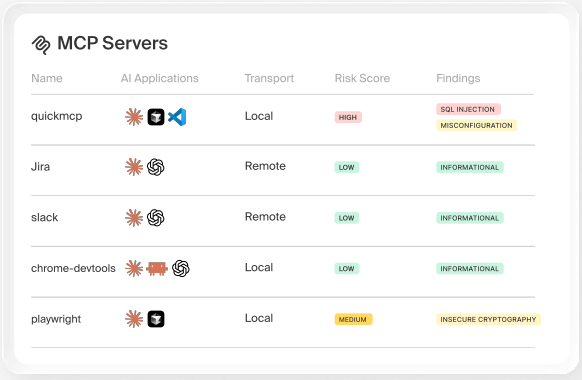


Agent Identity and Access Control

- An identity assigned to every agent with scoped permissions tied to the specific task, not granted broadly
- Non-human identity governance controlling which resources each agent can reach and under what conditions
- MCP Gateway governance brokering agent access to integrations with policy-driven permissions and a kill switch
- A handler model keeping a named human accountable over agents running at scale
- Full audit trails capturing every agent action and outcome

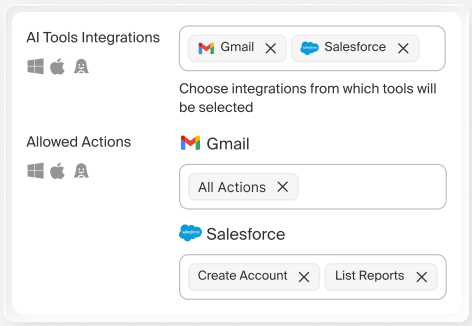
Threat Detection and Data Protection

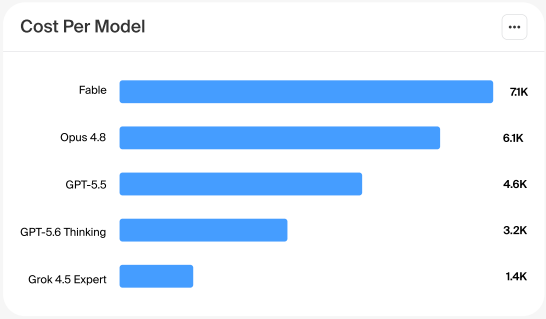
- Real-time classifiers that catch known threat types as they happen: prompt injection, jailbreak, agent drift, malicious URLs, malicious MCPs, data leakage, and unwanted content
- Two-layer data protection combining deterministic DLP with an evaluator that catches contextual and intent-level leakage
- Data boundaries that keep corporate data from moving to personal AI accounts or ungoverned environments
- AI-provider boundaries governing what data providers can see from application context
- The ability to prevent AI vendors from training their models on employee prompts
- In-browser guidance that redirects employees to approved tools and explains policy in context



AI Flexibility and Model Choice

- The ability to embed any AI provider the organization chooses, including Grok, ChatGPT, Gemini, Copilot, and Claude, with no vendor lock-in
- Smart model routing that directs the right AI to the right user based on role, task, and cost
- A production-ready enterprise chat interface supporting multiple frontier models simultaneously
- Support for an organization's own fine-tuned models alongside commercially available ones



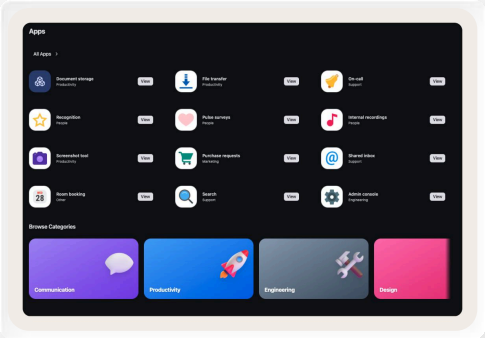


Cost and Experience Visibility

- Full visibility into token usage, model spend, and AI activity per user, team, and project
- Rate and budget controls at the user, team, or model level without cutting off access
- Agent experience monitoring: prompt latency, model response times, and tool-call performance so IT can see where agents are stalling or burning tokens
- The ability to right-size model selection to the task so the organization isn't paying frontier model prices for work a smaller model handles just as well
- Proof of which AI investments are paying back, giving the CIO and CFO the signal they need to scale with confidence

App Publishing and Governance

- Citizen-built apps automatically inherit SSO from the user's existing browser session
- Every published app sits inside existing DLP and application boundaries without additional configuration
- Apps inherit the organization's existing security posture without a per-app review process
- IT has full visibility into every published app, every user, and every interaction
- The organization can welcome apps at the speed employees build them without accepting ungoverned data exposure



The "Say Yes to AI" *Checklist*

Before security teams can confidently greenlight AI for the enterprise, certain controls must be in place. Use this checklist to evaluate your current readiness. The more items you can check off, the more safely your organization can move from AI experimentation to a workforce that's both human and agentic, operating at full scale.

Checklist:

AGENT VISIBILITY AND INVENTORY

- [ ] We have a full inventory of every agent running in our environment, regardless of vendor.
- [ ] We can see installed MCP servers, skills, code packages, browser extensions, and IDE extensions in real time.
- [ ] We have real-time visibility into what agents are actively doing, not just that they exist.
- [ ] We can detect unauthorized or unvetted agents before they reach internal systems or sensitive data.
- [ ] We have visibility into the full arc of multi-step agentic tasks as a single mission, not disconnected events.

AGENT IDENTITY AND ACCESS CONTROL

- [ ] Every agent has its own identity with scoped permissions tied to its specific task.
- [ ] We have a handler model in place with a named human accountable for agents running at scale.
- [ ] MCP server integrations used by agents are inventoried, risk-scored, and governed.
- [ ] We have a kill switch for MCP integrations when something goes wrong.
- [ ] Full audit trails capture every agent action and outcome across all surfaces.

THREAT DETECTION AND DATA PROTECTION

- [ ] Real-time classifiers catch prompt injection, jailbreak attempts, agent drift, and malicious MCPs as they happen.
- [ ] We have two-layer data protection combining deterministic DLP with contextual evaluation.
- [ ] Corporate data can't move to personal AI accounts or ungoverned environments.
- [ ] AI providers can only see what our policy allows them to see from application context.
- [ ] AI model training on employee prompts is disabled for all corporate accounts.

AI FLEXIBILITY AND MODEL CHOICE

- [ ] Employees can access multiple AI providers without being locked into a single vendor.
- [ ] Model routing is governed by policy based on role, task, and cost.
- [ ] We have visibility into which models users are accessing, how often, and at what cost.
- [ ] We support our own fine-tuned models alongside commercially available ones.

Checklist:

COST AND EXPERIENCE VISIBILITY

- [ ] Token usage and model spend are visible per user, team, and project.
- [ ] We have rate and budget controls at the user, team, or model level.
- [ ] We can see where agents are stalling, retrying, or burning tokens unnecessarily.
- [ ] We can right-size model selection to the task to avoid overspending on frontier models.
- [ ] We can demonstrate which AI investments are delivering measurable return.

APP PUBLISHING AND GOVERNANCE

- [ ] We know how many employee-built apps exist in our environment and where they're running.
- [ ] Citizen-built and vibe-coded apps automatically inherit SSO, DLP, and data protection when published.
- [ ] Published apps are visible to IT with full audit trails for every user and every interaction.
- [ ] Employees can access approved apps directly alongside the tools they already use.
- [ ] We're not dependent on engineering teams to make a user-built AI app enterprise-ready.

ORGANIZATIONAL READINESS

- [ ] We have a clear policy framework that covers all AI and agent entry points.
- [ ] Business units understand what AI tools and agents are approved and why.
- [ ] We have a process for evaluating and onboarding new AI tools and agents safely.
- [ ] We're not relying on 7 or more separate tools to achieve AI and agent coverage.
- [ ] The security team is positioned as an enabler of AI, not a blocker.

How to score yourself

Fewer than 20 checked: Your organization is in early-stage AI and agent security. Start with agent visibility and threat detection before expanding controls.

20 to 30 checked: You have partial coverage but meaningful gaps remain, particularly around agent identity and cost visibility. Prioritize those before scale compounds the risk.

31 to 40 checked: You have the controls in place to govern a workforce that's both human and agentic. Focus on compounding that advantage across the organization.

AI Maturity *Assessment*

Not every organization is at the same point in its AI security journey. Use this assessment to identify where your organization currently sits and what the most important next steps look like from there.

Stage 1 Reactive

You're aware AI and agent usage is happening but have limited visibility into what, where, or how. Security posture is largely reactive: block when something surfaces, investigate after incidents.

Key signals:

- No agent inventory
- Data protection relies on acceptable use policies, not technical controls
- No formal process for evaluating or approving AI tools or agents
- MCP integrations in use but unmonitored
- Agents have no identity, permissions, or named handler

Where to start:
Build your visibility foundation first. You can't govern what you can't see.

Stage 2 Developing

Some AI controls are in place but coverage is fragmented. Significant gaps remain across desktop, extensions, agents, and MCP integrations.

Key signals:

- Visibility exists for some AI tools but not others
- DLP covers web but not desktop AI
- Agents deployed without identity, permissions, or a handler model
- Prompt injection protections inconsistent across surfaces
- MCP servers lack inventory, risk scoring, or a kill switch
- Citizen-built apps being published outside the organization's visibility

Where to start:
Establish agent identity and a handler model before use scales. Stand up MCP governance. Get visibility into citizen-built apps before shadow app sprawl compounds your exposure.

Stage 3 Defined

Consistent AI controls exist across most entry points. Policies are documented and enforced. Gaps may remain in agent governance, cost visibility, and MCP oversight.

Key signals:

- Unified visibility across browser and desktop AI
- Data protection policies active and enforced
- AI access policies consistent across user roles
- Handler model emerging but cost governance and full audit coverage still maturing
- MCP governance in place but permission-combination detection still maturing
- Process exists for citizen-built apps but can't absorb the volume

Where to start:
Where to start: Build out cost and token visibility per agent, user, and team. Establish app publishing controls so citizen-built apps inherit security posture automatically.

Stage 4 Optimized

Full coverage across all AI and agent entry points. Controls are consistent, automated, and integrated with your broader security stack.

Key signals:

- Full agent inventory with real-time visibility into active behavior
- Agents run at scale under defined handlers with full cost accountability
- Real-time classifiers catch prompt injection, jailbreak, agent drift, and malicious MCPs
- Citizen-built apps published safely at scale with inherited controls
- Token costs and model spend visible and attributable across teams and projects

Where to focus next:
Your governance infrastructure is now the reason the organization can say yes to agents, citizen-built apps, and AI at scale. The security team isn't the blocker. It's the accelerator.

A One-Layer *Answer*

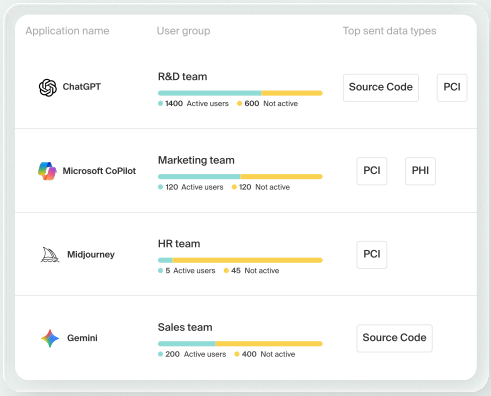
The controls described in this playbook are real. But most organizations trying to achieve them are doing so by assembling 7 or more point solutions across a fragmented entry point landscape, accepting coverage gaps and policy inconsistency as the cost of doing business.

Island is the Enterprise Agentic Control Plane: the single layer where both your people and your agents get what they need to work safely. A platform is where things are built or hosted. A control plane is where everything is governed. Island is the latter, and it's built specifically for a workforce that's both human and agentic.

Agentic Endpoint Posture

Island discovers every agent on the endpoint, regardless of vendor, and builds a full inventory: installed MCP servers, skills, code packages, browser extensions, IDE extensions, and the hooks that give real-time visibility into what's running. Most security tools know an agent exists. Island knows what the agent's *doing*.

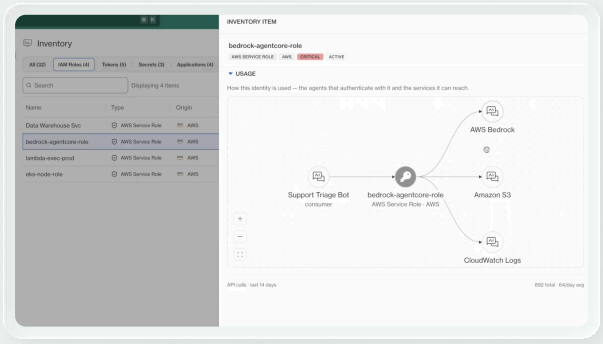
- Full agent inventory across every surface, updated in real time.
- Continuous behavioral monitoring so unauthorized or drifting agents are caught before they cause damage.
- Visibility into the full arc of multi-step agentic tasks as a single mission, not disconnected log entries across four tools.



Agentic Identity

Every agent that operates inside Island gets its own identity with scoped permissions tied to its specific task. Island's MCP Gateway brokers agent access to 500+ integrations with flexible, policy-driven permissions and a kill switch when something goes wrong. Agents don't get broad access by default. They get exactly what they need, under policy, with a full record of everything they did.

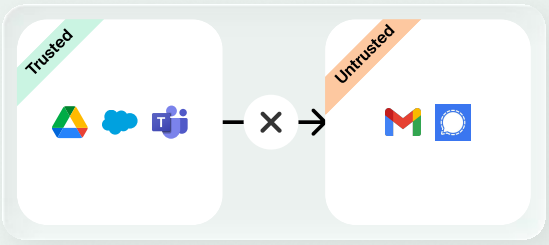
- Non-human identity governance that controls which resources each agent can reach and under what conditions.
- MCP Gateway with inventory, risk scoring, and a kill switch for every integration.
- A handler model with a named human accountable for agents running at scale.
- Full audit trails capturing every agent action and outcome across every surface.



AI Protect

Island sits inline between every agent and the enterprise, in real time, across browser, endpoint, and cloud. Real-time classifiers catch known threat types as they happen: prompt injection, jailbreak, agent drift, malicious URLs, malicious MCPs, data leakage, and unwanted content. A second evaluator layer catches the contextual and intent-level problems that pattern matching misses.

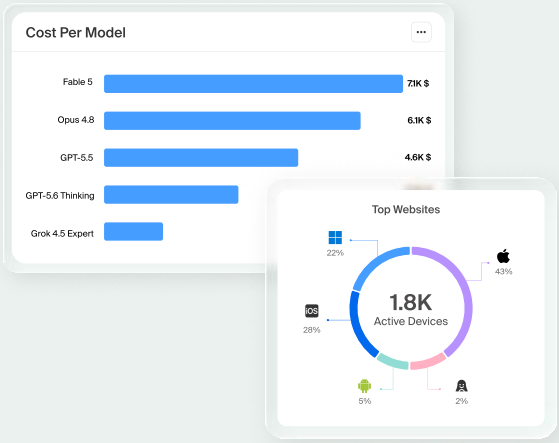
- Two-layer data protection combining deterministic DLP with contextual evaluation.
- Data boundaries that keep corporate data from moving to personal AI accounts or ungoverned environments.
- AI-provider boundaries governing what providers can see from application context.
- The ability to prevent AI vendors from training their models on employee prompts.



AI Cost and Experience

Island gives the CIO and CFO the signal they need to scale AI with confidence rather than hope. Every token, every model call, every agent interaction is tracked per user, team, and project. Rate and budget controls let organizations set limits without cutting off access. And digital experience monitoring extends to agents, so IT can see where agents are stalling, retrying, or burning tokens on slow MCP connections and overloaded models.

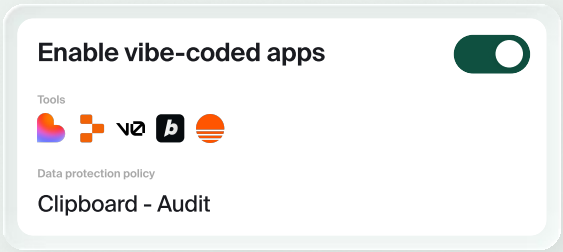
- Full token and spend visibility per user, team, and project.
- Rate and budget controls at the user, team, or model level.
- Agent experience monitoring: prompt latency, model response times, and tool-call performance in real time.
- Right-size model selection to the task so the organization isn't paying frontier model prices for work a smaller model handles just as well.



Enterprise Vibe Publishing

Employees are already building apps with Claude Code, Lovable, Cursor, and Replit at a scale enterprises have never seen. The building question is settled. The unsolved question is how the organization safely consumes them. Island Enterprise Vibe Publishing answers that. Apps live inside Island's existing controls the moment they're published, inheriting what would otherwise take months to build.

- SSO is inherited from the user's existing browser session, not built from scratch per app.
- Every app sits inside Island's existing DLP and application boundaries automatically.
- Governed private access connects each app to the specific internal resources it needs, with no broad network exposure.
- The organization's existing security posture extends to every published app without re-running architecture, compliance, or red-team review.



One control plane. Built for a workforce that's both human & agentic.

FINAL THOUGHT

AI security is not about saying "no."

Your workforce now includes agents. They work alongside your people, they move fast, and left ungoverned, they'll find ways to complete their mission that you didn't intend and can't see. The organizations that win with AI aren't the ones that blocked it longest. They're the ones that built the controls to govern it, deploy it safely, and scale it with confidence.

Security teams aren't standing in the way of AI. They're the ones who make it possible to go further, faster, and with confidence.

Island is the Enterprise Agentic Control Plane: the one place you onboard, enable, govern, and account for your agentic workforce, alongside the people they work with. One control plane. Built for a workforce that's both human and agentic.

Ready to say "yes" to AI?

Schedule a demo | island.io ↪

Island