SACR

OCT 2026

THE CISO'S GUIDE TO CLOUD SECURITY INTENT ENFORCEMENT

JAMIE FEDER

PAUL WEBBER



native

We explore the newest frontiers of cybersecurity.

Whether you're looking at emerging vendors, evolving threats, or shifting architectures, our timely, opinionated insights help modern security leaders make smarter, faster decisions.

About Us

Software Analyst Cyber Research (SACR) is a modern research and advisory firm built for today's cybersecurity leaders. We deliver in-depth, timely analysis across SOC operations, Identity, Network, Cloud, Application Security, Data, and AI Security; equipping CISOs, security teams, founders, investors, and practitioners with the insight they need to navigate high-stakes decisions.

With an engaged community of over **80,000 readers and followers**, SACR connects with a global network of cybersecurity decision-makers and innovators. Our access to leaders across categories and industries gives us a direct line to the conversations shaping the market. By pairing these insights with rigorous technical analysis and continuous market tracking, we produce research that is both data-driven and grounded in the realities of modern security operations.

Whether you're seeking clarity on emerging technologies, evaluating vendors, or tracking market shifts, SACR delivers trusted, independent research designed to help you see clearly and decide with confidence.

The Authors

Jamie Feder serves as an Associate Research Analyst at SACR, where she researches cloud security, identity, and the security implications of AI and emerging technologies. A Princeton University graduate, she holds a master's degree in Security and Diplomacy from Tel Aviv University. Based in Tel Aviv, she translates complex cybersecurity developments into clear research and analysis, helping security leaders understand evolving risks, assess vendor approaches, and evaluate new security frameworks.

Paul Webber has over 28 years of experience in cybersecurity and enterprise technology, with expertise spanning cybersecurity strategy, threat detection and response, SOC operations, cloud transformation, and cyber resilience across enterprise environments. A former Gartner Senior Director, he advised global CISOs and security vendors on product strategy, go to market alignment, and emerging cybersecurity trends. He brings extensive practitioner experience, advising enterprise security leaders on emerging security technologies, operational strategy, ransomware defense, and the evolving cyber threat landscape.

TABLE OF CONTENTS

Executive Summary 3

Defining Cloud Security Intent Enforcement (CSIE) 4

A Brief History 6

How CSIE Complements CNAPP 7

The CSIE Control Loop 9

Innovative Capabilities Emerging Across CSIE 11

How SACR Evaluates CSIE 12

Exception and Policy Operations 14

Keeping Controls Up to Date 14

AI Agents and Environmental Boundaries 15

Buyer Guidance 16

Native 23

Sponsorship and Research Disclosure 28

Conclusion 29

Executive Summary

Cloud security teams already have extensive visibility into risk, and often know which cloud states they do not want to allow. The harder problem is preventing those states from being introduced or recreated across constantly changing cloud environments, without slowing the business down or *causing unacceptable production disruption*. The proliferation of artificial intelligence (AI) and agents only exacerbates *the need for comprehensive proactive solutions*, rather than just reactive responses. Google Cloud's *H1 2026 Threat Horizons report* calls for stronger identity controls and automated posture enforcement as cloud threats accelerate.

Cloud Security Intent Enforcement (CSIE) addresses the gap between knowing what should be secure and ensuring it stays that way. CSIE connects cloud context and security intent with impact analysis, preventive enforcement, and ongoing control management to help organizations prevent unsafe conditions from recurring.

An emerging group of specialist vendors, including Act Security, Native, and Aryon Security, is approaching the problem through intent-driven cloud controls. At the same time, incumbent cloud-native application protection platform (CNAPP) vendors are beginning to absorb parts of the CSIE control loop.

Wiz, in particular, is extending its existing cloud context into preventive policy enforcement, impact-aware remediation, and developing cross-cloud policy workflows. This raises a broader market question: Will CSIE develop into a distinct security layer, or will its capabilities become part of the platforms organizations already use?

For CISOs, the question is not whether a vendor can call itself a proactive vs a reactive approach, but whether the platform can turn a security objective into a durable control that is enforced without creating unacceptable, and often unpredictable, operational risk. CSIE should ultimately be judged by that outcome, not by adherence to a fixed feature checklist.

Why Now?

Recent research and cloud attacks underscore the need for proactive security enforcement. The *2026 SANS Cloud Security Survey* found that 45% of respondents identified misconfiguration as the leading cause of critical cloud incidents, while 49% took more than 24 hours to close a known critical gap. In September 2026, *Microsoft documented an Azure attack* involving more than 100 attempted storage account deletions in approximately seven minutes, some of which were blocked by preventive safeguards. Meanwhile, the *Cloud Security Alliance's 2026 threat report* ranked inadequate identity and access management as the leading cloud security threat. Together, these developments reinforce the need for durable cloud controls that prevent unsafe conditions rather than relying solely on detection and remediation.

Defining Cloud Security Intent Enforcement (CSIE)

SACR defines CSIE as the control layer that takes cloud context and declared or derived security intent, translates that intent into enforceable controls, evaluates the likely impact of the change, applies the control through an appropriate enforcement point, and maintains the intended state over time. The objective is not simply faster remediation, but reducing the need to repeatedly rediscover and fix the same unsafe conditions.

Proactive enforcement is a defining requirement for CSIE. In other words, a qualifying product must materially participate in creating and operationalizing a control that prevents a prohibited cloud state before it becomes active. As SACR defines it, the product does not need to write directly into production, provided the resulting control reaches an authoritative enforcement point and its effectiveness can be verified.

CSIE does not imply that proactive cloud controls are new, nor that every control must be applied before deployment. Indeed, cloud providers, CNAPP platforms, identity tools, and security vendors already support different forms of proactive enforcement.

What is changing is the way these capabilities are being combined in a more continuous workflow with greater predictability, connecting **context, intent, impact analysis, enforcement, and ongoing policy operations**. Existing frameworks such as the *CSA Cloud Controls Matrix v4.1* already establish cloud security control objectives. CSIE focuses on how organizations translate those requirements into enforceable controls and maintain them over time.

The following market map illustrates overlapping cloud security approaches and capabilities, with representative vendors appearing across multiple areas based on their demonstrated functionality.



Figure 1. CSIE Market Map

A Brief History: From Visibility to Control

Cloud security began as a fairly simple endeavor: find insecure cloud configurations. Early cloud security posture management (CSPM) tools helped teams identify exposed resources, weak settings, and compliance issues. As cloud environments grew, security products expanded into identity, entitlements, workloads, vulnerabilities, data, and runtime activity. (For a fuller history, see *SACR's earlier Agentic Cloud Security Platforms (ACSP) research*, which traces the evolution through CNAPP and toward runtime control.)

Many of these capabilities eventually converged in CNAPP. The improvement was not just broader coverage: CNAPP platforms began combining findings with context.

A vulnerability on an isolated internal system, for example, presents a different risk from the same vulnerability on an internet-facing workload that can reach sensitive data through a privileged identity. ACSP extended this evolution by asking whether a workload, identity action, or agent-driven operation should proceed while it is in motion.

CSIE is not CNAPP 2.0, an evolution of CNAPP, or a replacement for existing posture, detection, or runtime platforms. Rather, CSIE addresses a related but more specific control problem: how to turn an organization's intended security boundary into a durable cloud control, understand the likely impact of applying it, introduce it safely, and keep it effective as the environment changes.

The challenge is not simply identifying or remediating a risk once. Another API call, console change, or deployment pipeline can recreate the same condition. CSIE therefore focuses on making the intended state harder to violate in the first place, without creating unacceptable disruption to the business.

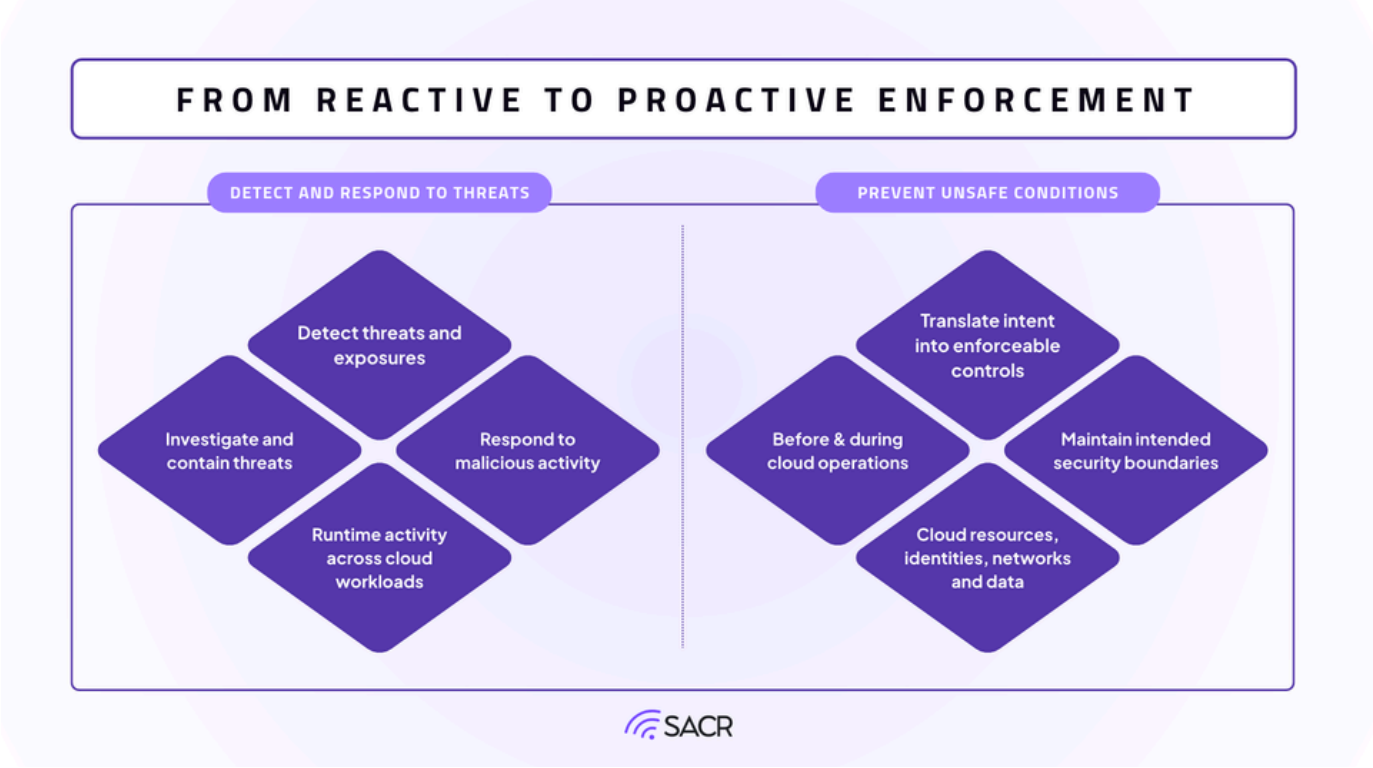


Figure 2: From Reactive to Proactive Enforcement

How CSIE Complements CNAPP

CSIE is not a replacement for CNAPP, but a complementary control layer that builds on it. In many environments, CNAPP already provides much of the context needed for prevention, including findings, identities, vulnerabilities, exposed resources, attack paths, and runtime signals.

Cloud hardening is closely related to this process, focusing on reducing unnecessary access, tightening configurations, and removing exploitable paths. CSIE builds on these practices by helping organizations turn security requirements into proactive controls that can be maintained as cloud environments change.

Building on this context, CSIE addresses a different set of questions:

- Which states and actions should the cloud permit?
- What control would prevent an unsafe condition from occurring or recurring?
- What legitimate resources and workflows would that control affect?
- Can it be introduced without unacceptable disruption?
- Can the organization keep it effective as the environment changes?

For specialists, the challenge is demonstrating enough additional depth in intent translation, impact prediction, enforcement, and ongoing policy operations to justify a separate product.

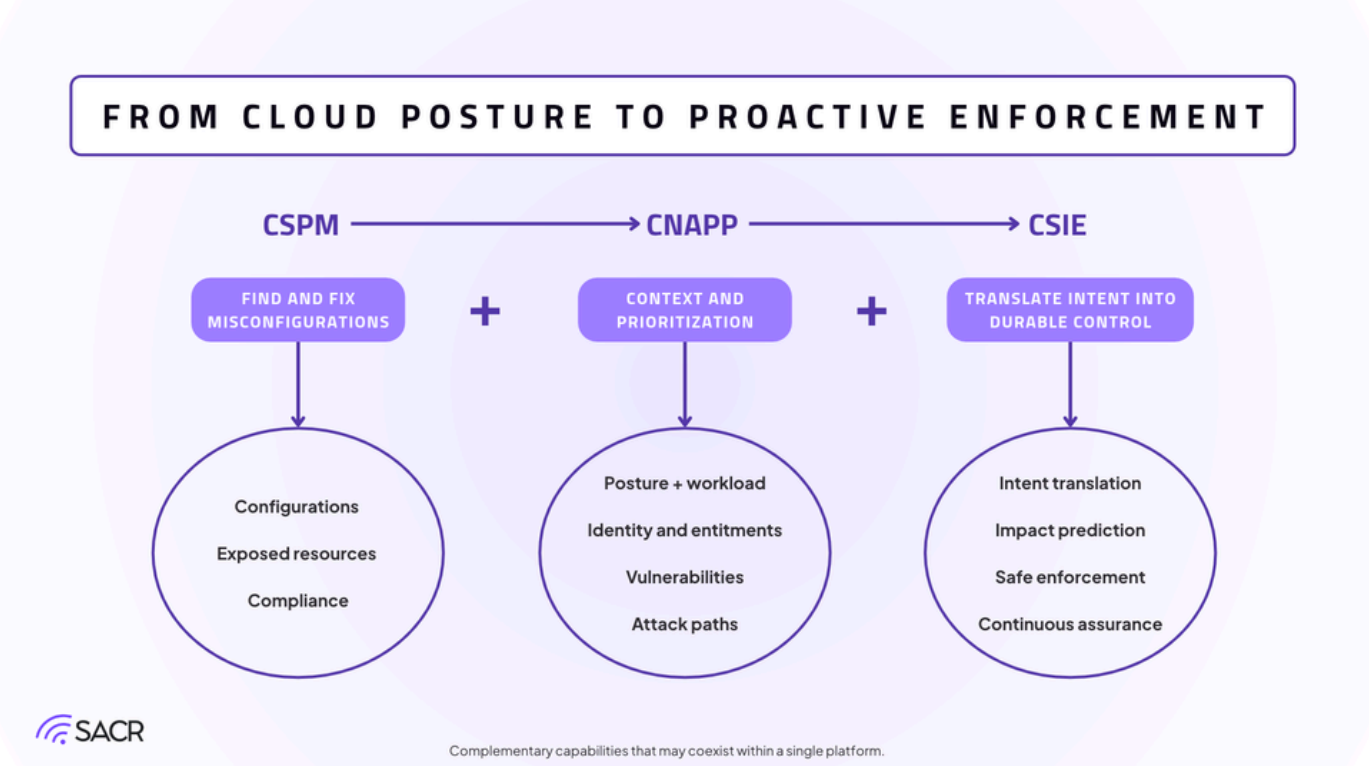


Figure 3. From Cloud Posture to Proactive Enforcement

The CSIE Control Loop

The cleanest way to think about CSIE is as a **continuous control loop**, not a checklist of mandatory features. The challenge is not simply to enforce more policy, but to introduce proactive controls without breaking legitimate applications, identities, network paths, or deployment workflows. That makes impact analysis, staged rollout, exception handling, and rollback central to safe enforcement in production.

The process begins by establishing the context needed to define security intent: access intent draws on identities, applications, agents, data, and configured or effective reachability, while posture intent may begin with a required configuration, benchmark, or policy state. The platform then translates that intent into cloud-specific controls for enforcement.

Before broad enforcement, the team needs credible assurance about likely operational impact: what risk will be removed, which resources or workflows could be affected, where dependencies exist, and where the analysis may be incomplete. Depending on the product, that assurance may come from simulation, historical activity, audit or alert mode, staged scope, dependency analysis, policy testing, rollback validation, or another demonstrated mechanism.

Once approved, the control must reach an authoritative enforcement point that changes what the cloud environment permits. This may occur through provider-native controls, direct cloud enforcement, or an authoritative customer workflow such as Infrastructure-as-Code.

This maps to the five-stage CSIE framework: **Understand, Translate, Predict, Enforce Safely, and Govern and Maintain**.

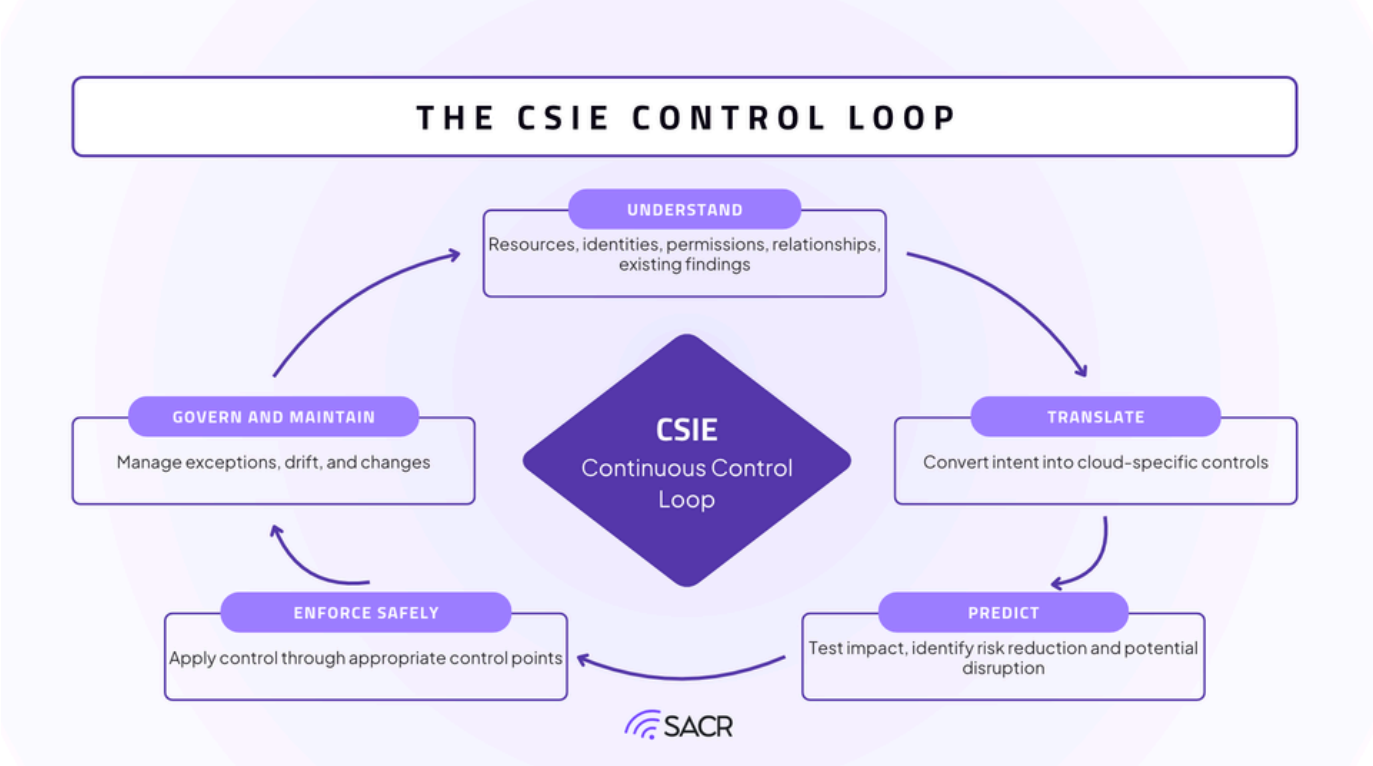


Figure 4. The CSIE Control Loop

The relative weight of each stage varies by control type. Access enforcement generally requires deeper environmental and dependency context, while posture enforcement may begin from a more declarative policy or benchmark.

Context completeness should be assessed against the scope of the control being enforced, rather than assuming full visibility across the cloud environment. Buyers should evaluate coverage of in-scope resources, identities, permissions, access paths, and dependencies, while identifying gaps in visibility that could affect impact prediction or enforcement.

A qualifying CSIE product must materially participate in creating and operationalizing a proactive control, with sufficient contextual information and production-safety assurance for that control to be applied responsibly. The framework then evaluates how deeply the product supports intent translation, impact prediction, enforcement, governance, and outcome verification.

Discovery, prioritization, recommendations, tickets, or generated fixes may support CSIE, but do not qualify on their own unless the resulting proactive control reaches an authoritative enforcement point. Likewise, detecting an unsafe state after creation and automatically remediating it does not, by itself, constitute CSIE; the workflow must prevent the prohibited state at an authoritative enforcement point.

Innovative Capabilities Emerging Across CSIE

The most important development in CSIE is not a single new feature, but the way several previously separate cloud-security capabilities are beginning to converge around proactive control. Across the vendors in this report, three technical patterns stand out.

- **Production-safety assurance** is becoming a major point of differentiation. Vendors use differing methods, including simulation, historical activity, audit modes, staged rollout, dependency analysis, and rollback validation, to understand what a proposed control may affect before broad enforcement. These approaches vary significantly in depth, and buyers should evaluate both what each method can predict, and where its blind spots remain.
- **Native control orchestration** is another important pattern. Rather than introducing a new inline enforcement layer, several vendors rely on controls already available within cloud providers or established Infrastructure-as-Code workflows. This shifts differentiation away from the enforcement primitive itself and toward how well the platform translates intent, selects the appropriate control, integrates with existing change processes, and verifies that the intended state is actually in effect.
- **Governed enforcement** may ultimately matter as much as the control itself. Staged rollout, exceptions, approvals, rollback, drift management, and ongoing verification determine whether proactive controls can operate safely at enterprise scale. Without those capabilities, even a technically correct policy may be difficult to trust in production.

How SACR Evaluates CSIE

The five stages of the CSIE control loop provide a common framework for understanding how proactive cloud controls are developed, introduced, and maintained. For vendor evaluation, SACR consolidates these stages into three capability areas: **Intent and Prevention, Assurance and Safety, and Governance and Continuity**. These are not separate frameworks, but complementary ways of assessing how effectively a product supports the control lifecycle.

The evaluation focuses on demonstrated capabilities rather than the presence of individual features. Vendors may address different control problems and rely on different technical approaches, but the underlying questions remain consistent: Can the product translate security intent into durable prevention?

Can those controls be introduced with sufficient assurance about their operational impact? And can the organization maintain them as cloud environments change? Customer outcomes are considered separately, with particular attention to risk reduction, operational efficiency, and the ability to enforce security boundaries without disrupting legitimate activity.

1. Intent and Prevention

The platform can convert cloud context and declared or derived security intent into a proactive control across the change paths that matter. This includes environmental and access context, intent translation, material participation in proactive enforcement, and coverage across console, CLI or PowerShell, APIs, IaC, CI/CD, scripts and automation, third-party systems, and AI or coding agents. Evidence from one change source should not be treated as proof of source-independent enforcement.

2. Assurance and Safety

The platform can provide credible assurance about likely operational impact before broad enforcement, stage the change safely, identify known limitations, and recover when assumptions are wrong. Relevant evidence may include simulation, historical replay, audit or alert mode, staged scope, dependency analysis, rollback readiness, and verification after enforcement that the control is operating as expected.

3. Governance and Continuity

The controls can actually be operated inside a large enterprise. This includes exceptions, approvals, separation of duties, consistent security intent across clouds, enforcement depth by provider, and the ability to prevent or correct drift over time.

Outcomes

The capability areas describe how CSIE works; outcomes describe what the customer should actually gain from it. Relevant outcomes include structural risk reduction, preventing recurring unsafe states, reducing exploitable or unnecessary access, improving cloud-operations efficiency, supporting compliance objectives, and enabling AI or automation to operate within enforceable boundaries.

These areas should guide the research without turning the report into a vendor scorecard. The current framework is still intended to test the market definition and boundaries, including whether customers recognize the problem and whether the controls can be demonstrated in production rather than merely stated on a roadmap.

Exception and Policy Operations

Exception handling is one of the capabilities that determines whether prevention can work safely in a real enterprise.

- For instance, if a policy blocks a legitimate action, the engineer needs to understand what happened and what to do next. The platform should identify the applicable policy, explain why the action was blocked, and provide a clear route for correcting the action or requesting an exception when appropriate.
- Exceptions should have an owner, justification, defined scope, approval, disposition, and complete audit history, with review or expiration where appropriate. Known legitimate exceptions should be identified before broad enforcement where possible, while new exceptions should support clear renewal and revocation workflows rather than becoming permanent bypasses. Emergency overrides may still be necessary, but they should be time-bounded, attributable, and reviewed after use.
- Security teams should be able to identify exceptions that are repeatedly renewed, used outside their intended scope, or accumulating to the point that the original policy no longer reflects reality.

For CISOs, the question is not simply whether the platform supports exceptions, but whether it provides enough governance to operate proactive controls at scale.

Keeping Controls Up to Date

Proactive controls also have to survive changes in the environment. Cloud services, APIs, permissions, workloads, and organizational structures all evolve, creating the risk that a once-effective control becomes incomplete or outdated.

A CSIE platform should therefore be able to identify drift, verify that intended controls remain active, and adapt policies when the underlying environment changes. Buyers should also understand what happens if the platform itself becomes unavailable and whether existing controls continue to operate independently.

This aligns with *NIST SP 800-37 Rev. 2*, which emphasizes continuous monitoring, control assessment, and accountability throughout the system lifecycle.

The most useful measures are operational: how quickly a control can be updated, how often policies require rollback, how many exceptions accumulate, and how consistently intended controls remain active over time.

AI Agents and Environmental Boundaries

AI and coding agents increase the importance of enforceable cloud boundaries without fundamentally changing the CSIE model. Because AI and coding agents can inherit existing human or service identities, they can exercise overly broad access at greater speed and scale. This raises the importance of durable identity, network, and resource boundaries that constrain what those agents can reach and do. This can also make AI adoption a buying trigger for CSIE, as organizations look for enforceable boundaries around the cloud resources, services, and actions available to agents. *OWASP's Excessive Agency guidance* identifies excessive permissions, functionality, and autonomy as core risks in agentic systems, reinforcing the need to restrict what agents can access and execute.

- Rather than trying to predict every action an agent may take, organizations can restrict the environment in which it operates: which resources it can reach, which actions it can perform, and which cloud services, models, or datasets it may use.
- The important distinction is enforcement. Instructions inside an application or agent are not equivalent to a cloud control. CSIE becomes relevant when those boundaries are enforced through the underlying environment and apply to agent-driven activity through APIs and automation.

This complements identity security rather than replacing it. Identity tools determine who or what is acting; CSIE helps constrain what that identity can cause the cloud environment to permit.

Where CSIE Is Most Useful

CSIE is most relevant in mature cloud environments where the organization already understands the outcomes it wants to enforce but struggles to apply them consistently.

- The strongest candidates are large engineering organizations, multicloud enterprises, regulated companies, and teams repeatedly encountering the same classes of cloud risk. These environments have enough complexity that manual remediation and one-off controls become difficult to sustain.
- Regulated organizations see particular value because some failures cannot safely exist even temporarily. Preventing an unauthorized exposure or prohibited production change can be far more valuable than detecting it quickly after the fact.

This model is less compelling for organizations that still lack basic inventory, posture visibility, or identity context. In those environments, the immediate problem is still understanding what exists and where the risk is; proactive enforcement becomes more useful once the organization knows which boundaries it actually wants to maintain.

Buyer Guidance: What Outcomes Should CSIE Deliver?

Customer-Reported CSIE Use Cases

20 practitioner-reported use cases across eight themes. Evidence note: Customer-reported, anonymized capabilities and operational outcomes. Not independently audited, guaranteed for every deployment, or evidence of quantified risk reduction.

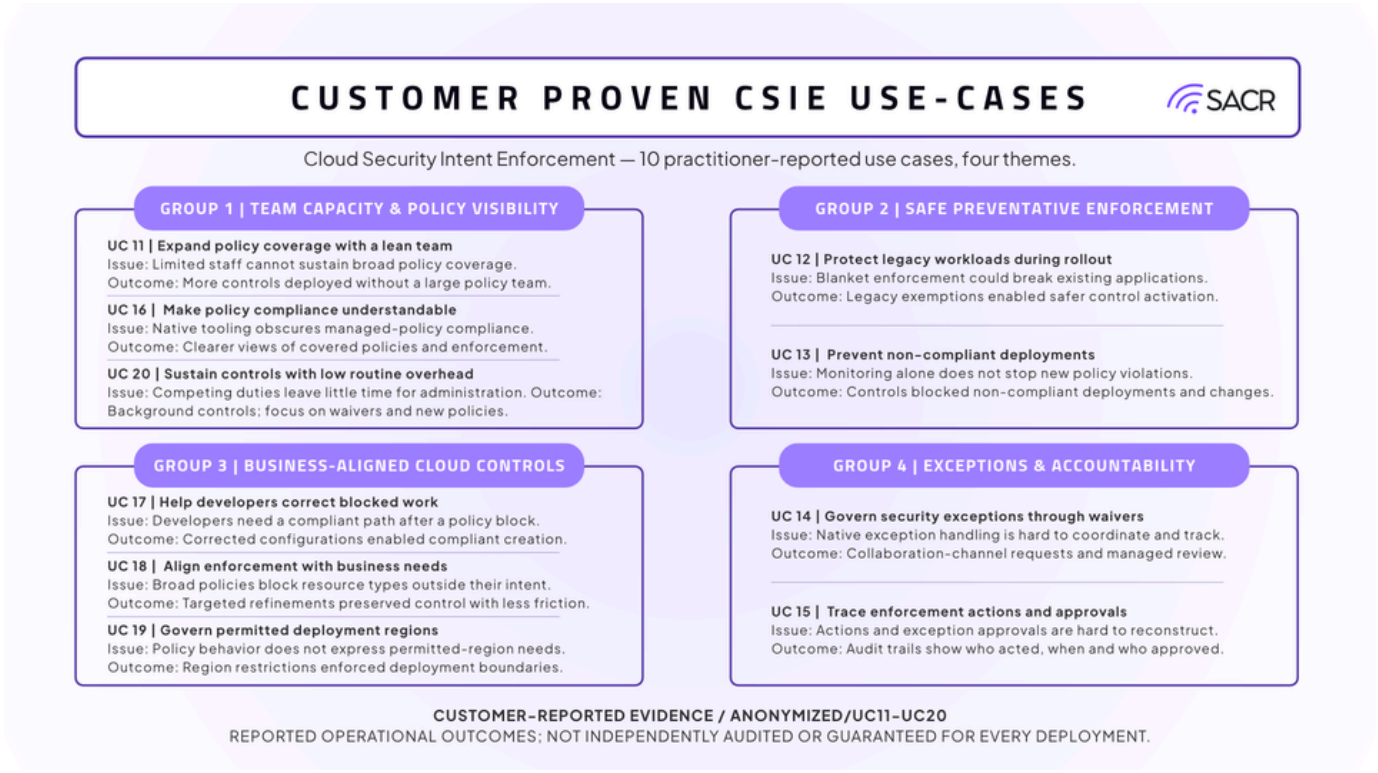


Figure 5: Customer Proven CSIE Use-Cases

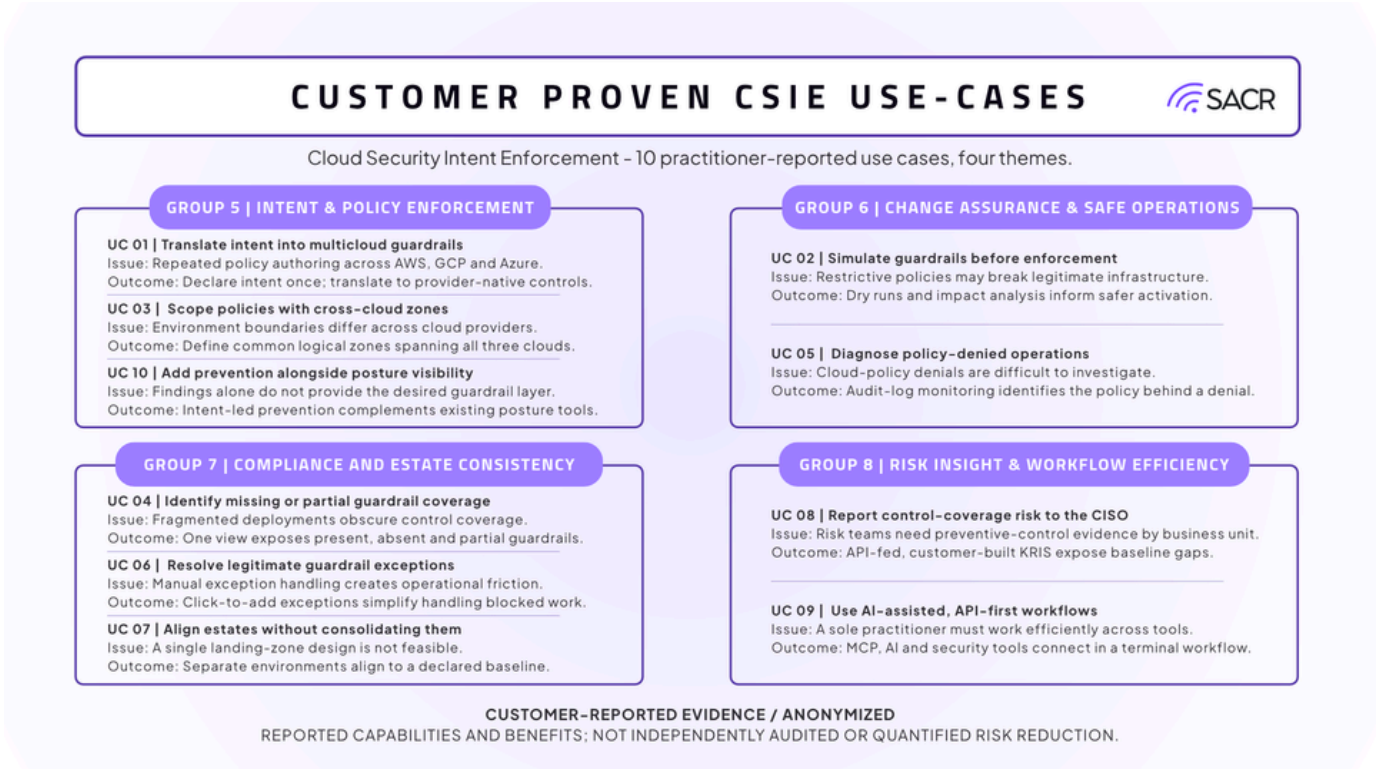


Figure 6: Customer Proven CSIE Use-Cases

For buyers, the most useful way to evaluate CSIE is by whether it materially improves security outcomes while reducing the operational burden on already stretched cloud security teams.

A strong CSIE deployment should help the organization **prevent unsafe cloud changes before they become another finding to investigate**. That might mean blocking a prohibited identity and access management (IAM) change, stopping a resource from being created in an unapproved state, or preventing a configuration change that would expose sensitive infrastructure. The key is that the control interrupts the unsafe state rather than reporting it afterward.

It should also help **reduce exploitable attack paths**. If a security team already knows that a combination of network access, permissions, and resource relationships creates a route to a sensitive asset, CSIE should help remove or prevent the relationships that make that path possible. CNAPP identifies the route; CSIE should help change the environment so it no longer exists or cannot be recreated.

A third outcome is **limiting identity and resource blast radius**. CSIE can help enforce boundaries around which identities may reach which resources, what actions they may take, and how far a compromised identity, workload, or resource can affect the rest of the environment. *MITRE ATT&CK's Valid Accounts: Cloud Accounts* documents how adversaries abuse legitimate cloud identities for initial access, persistence, and privilege escalation, highlighting the importance of limiting unnecessary permissions and access paths.

The platform should also make it easier to **enforce architectural intent consistently**. Organizations already know that production and development should remain separated, sensitive data should only be accessible from approved systems, and certain identities should never receive broad privileges. CSIE should help translate those principles into controls that remain consistent across accounts, providers, teams, and change paths.

Engineering teams should be able to move faster without adding security risk. If the platform can predict the impact of a control, explain why an action is blocked, automate routine enforcement, and provide a clear exception process, security teams can enforce stronger boundaries without reviewing every individual change.

That gives buyers a concise set of outcomes to test:

- Prevent unsafe cloud states before they become a persistent risk.
- Reduce exploitable attack paths and unnecessary access to critical systems and data.
- Limit identity and resource blast radius.
- Enforce architectural intent consistently across clouds, accounts, teams, and change paths.
- Reduce recurring remediation and cloud-security operating effort.
- Let engineering teams move faster within defined security boundaries.

Not every buyer will weigh these outcomes equally. A regulated financial institution may care most about preventing prohibited states from existing at all, while a large engineering organization may place more value on reducing manual approvals and letting developers work within predefined boundaries. Buyers should start with the outcome they need to improve and work backward into the controls required to achieve it.

Buyer Decision Guide

Not every organization needs the same type of CSIE capability. The most useful starting point is the control problem the buyer is trying to solve, rather than the feature list of an individual vendor.

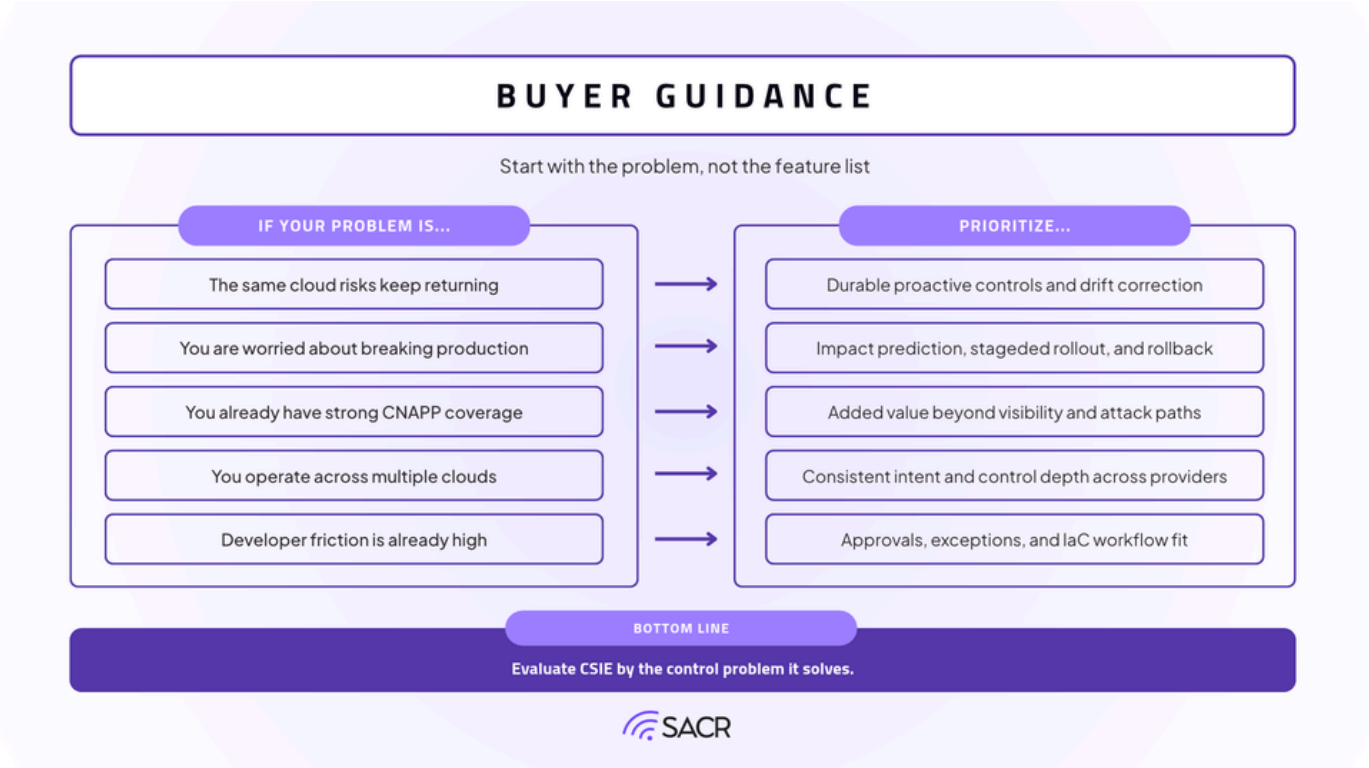


Figure 7: Buyer Guidance

The evaluation should then focus on which parts of the CSIE loop matter most for that problem: durable prevention, impact prediction, exception handling, multi-cloud consistency, developer workflow integration, or coverage across machine-driven change paths.

What Buyers Should Ask Vendors to Prove

The vendor should be able to show more than a clean demo of a policy being blocked. A buyer should ask the vendor to begin with a real environment or representative architecture, identify an actual security outcome, translate that outcome into a control, show what the control is expected to affect, and then demonstrate what happens when the policy is introduced.

The evaluation should include both the successful case and the failure case. What happens when legitimate activity would be blocked? Can the product identify the affected workload or owner? Can the control be staged rather than applied everywhere at once? Can it be rolled back quickly? Can the team see exactly what changed?

Buyers should also test how the platform behaves outside the vendor's ideal workflow. If a policy is enforced through CI/CD, what happens when a change is made through the console or API? If it works well in Amazon Web Services (AWS), how much functionality is preserved in Azure or Google Cloud Platform (GCP)? If a developer receives an exception, does that exception expire automatically or remain indefinitely? Evidence from one change source should not be treated as proof that the same control applies across console, API, CLI, IaC, automation, and agent-driven changes.

Customers should ask for measurable production evidence. Relevant outcomes might include the number of unsafe changes prevented, attack paths removed, repeated findings eliminated, time saved in remediation, reduction in exception volume, policy rollback rates, or the amount of manual engineering work avoided.

A vendor may have an impressive architecture without yet having strong evidence across all of these areas. That is not necessarily disqualifying in an emerging market, but the report should distinguish clearly between what the product can technically do and what customers have demonstrated at scale.

Will CSIE capabilities be introduced by incumbent CNAPP solutions?

Established CNAPP platforms have a structural advantage in CSIE because they already own much of the context required to make preventive controls useful. Wiz, for example, connects resources, identities, permissions, vulnerabilities, data, code, and runtime signals through its Security Graph, giving it a strong foundation for understanding where a proposed control should apply and what it may affect.

More importantly, Wiz is beginning to extend that context into parts of the CSIE control loop itself. Its unified policy engine can apply preventive guardrails through development and deployment workflows, while Green Agent uses Graph, ownership, configuration, historical activity, and code-to-cloud context to evaluate remediation paths and move changes into operational workflows.

This means the incumbent question is no longer simply whether CNAPP platforms could move from visibility into control. Wiz already demonstrates meaningful capability across context, preventive enforcement, and impact-aware remediation.

The remaining question is depth: whether an organization can begin with a new architectural requirement, translate it consistently into authoritative IAM, network, resource, or organizational controls across providers, predict the operational consequences of those controls, and govern them over time through exceptions, rollback, drift management, and verification.

The competitive tension therefore works both ways. Incumbents need to demonstrate that these capabilities form a durable end-to-end control loop rather than a collection of preventive and remediation features. Specialists need to show that their deeper intent translation, change assurance, enforcement, and policy operations provide enough incremental value to justify another product in the stack.

How Vendors Map to CSIE

CSIE vendors approach the control loop from different starting points. Act Security, Native, and Aryon Security represent the clearest specialist approaches reviewed here, with emphasis respectively on architecture and access boundaries, intent-to-control translation, and change-path enforcement and governance.

Incumbents also matter. Wiz and Orca already own much of the cloud context on which CSIE depends. Their next steps are important to watch as this category grows increasingly prescient.


native

Vendor Profile

Native is a multicloud security control platform that translates security requirements into provider-native controls across AWS, Microsoft Azure, Google Cloud, and Oracle Cloud Infrastructure (OCI). Founded in 2024 by Amit Megiddo, Gal Ordo, and Eyal Faingold, *Native has raised $42 million*. Megiddo previously led Amazon GuardDuty, Ordo led AWS Security Hub, and Faingold previously served as VP of R&D at Dome9 and later VP of Cloud Security at Check Point.

Cloud security increasingly depends on architecture that can enforce what should and should not be possible, rather than relying only on detecting risk after it appears. The cloud providers already expose the controls needed to create those boundaries, but they are fragmented across services, policy models, and providers and are difficult to operate consistently at enterprise scale. Native turns those built-in controls into operationalized secure architecture across AWS, Azure, Google Cloud, and OCI.

While there is no dependence on a CNAPP solution, the Native platform adds an additional layer of capabilities: Where CNAPP platforms typically provide broad visibility across posture, vulnerabilities, identities, exposures, and attack paths, Native focuses on translating selected risks and architectural requirements into enforceable control, including requirements that do not originate from an existing finding or exposure.

Architecture and Core Functions

Native builds an architectural model of the cloud environment across resources, identities, accounts, relationships, access patterns, and existing controls. Its architectural graph, deployment zones, security baselines, and control mappings help determine where a security requirement should apply and how that requirement should be implemented within each cloud provider.

Teams can begin by declaring a desired security or architectural outcome, including requirements that may not be represented in the environment's current configuration or historical access patterns. Native then uses its model of the environment to determine where that requirement applies and how it should be implemented. These requirements can span perimeter controls, segmentation, identity, data access, network architecture, regional and service restrictions, security baselines, and boundaries governing (as an example) what AI agents can access or do.

This context can also incorporate findings from CNAPP platforms. A recurring exposure or configuration issue identified elsewhere can become an input into a broader preventive policy where appropriate.

Simulation is a central part of the workflow. Before deployment, Native evaluates a proposed policy against the modeled environment and, where supported, historical cloud activity to identify resources, identities, access patterns, and workflows that could be affected. Teams can review expected impact, exclude specific resources, and create scoped or time-bound exceptions before enforcement. Native recalculates the expected result as those exclusions change and can also support post-deployment monitoring and rollback rules tied to operational indicators such as error-rate thresholds.

Once approved, Native translates the desired outcome into controls enforced by the underlying cloud provider. Depending on the provider and use case, these may include organization policies, service or resource control policies, IAM and role-based access control (RBAC) controls, resource policies, and network restrictions. Native does not operate as an inline proxy; enforcement remains with the cloud provider.

The same model can be used to establish boundaries for AI agents operating in cloud environments, including which services, resources, data, regions, or actions the cloud should allow. Enforcement remains in the cloud provider's native control layer rather than in the agent itself.

Native also continues to manage controls after deployment. The platform tracks exceptions, identifies drift, monitors deployed policies, and updates its control logic as cloud-provider services and APIs change. When an action is blocked, affected users can be notified and routed into an exception workflow, while modified or removed provider-native controls can be identified and, where configured, restored or updated.

Differentiation

Native's differentiation comes from combining declared security intent, cross-cloud translation, impact simulation, enforcement, exception handling, and ongoing alignment within one workflow. This allows teams to define their desired policy and configuration architecturally, rather than relying only on existing configuration, discovered exposures, or observed access patterns to determine the desired state.

Cloud providers already supply extensive enforcement mechanisms, while infrastructure-as-code provides a mature way to deploy known configurations. Native operates above those mechanisms by helping teams determine how a common security requirement should be implemented across different providers and then managing that control over time.

The distinction from CNAPP is similarly complementary. Whereas, a typical CNAPP provides the visibility and risk context needed to understand where security problems exist, Native focuses on architectural requirements and turning organizational policies into controls that can be enforced and maintained across cloud environments, defining the security and architectural state an organization intends to maintain, and managing those controls and exceptions over time.

Areas to Validate

Native's core workflow across intent translation, simulation, provider-native enforcement, exception management, and drift detection is supported by SACR's current research. The main areas to validate are provider depth, simulation reliability, operational efficiency, and the consistency of outcomes across different environments.

Provider support should not be assumed to be identical across AWS, Azure, Google Cloud, and OCI. Buyers should test the specific controls and services relevant to their environment and understand which policies can be modeled using configuration data, historical activity, architectural context, or some combination of those inputs.

Simulation should also be treated as a risk-reduction mechanism rather than a guarantee against disruption. Infrequent dependencies, undocumented workflows, and internal cloud-provider behavior may not always be visible in advance.

SACR has received customer validation for Native's cross-cloud intent translation, simulation, exception handling, coverage reporting, and complementary use alongside CNAPP. One customer estimated upwards of 50% time savings for a specific "no new static credentials" use case across three cloud providers.

SACR Perspective

Native is most relevant to larger, cloud-mature organizations operating across multiple providers, accounts, subscriptions, projects, or business units where security teams need to apply common architectural and security requirements consistently.

It is particularly relevant to organizations that already use CNAPP for visibility and prioritization but want a more centralized way to translate those decisions into provider-native controls, understand likely impact before deployment, and manage exceptions and drift afterward.

Smaller or primarily single-cloud organizations should compare Native with the policy-management capabilities already available through their cloud provider and existing IaC processes. Native becomes more relevant as provider diversity, policy complexity, and the ongoing burden of maintaining consistent controls increase.

Evidence Basis

SACR evaluated vendors against the CSIE control loop using product demonstrations, vendor briefings, available customer and practitioner evidence, public product documentation, and observed workflows. Capabilities are credited based on evidence reviewed to date, with generally available and demonstrated functionality distinguished from vendor claims, roadmap items, and areas that still require further validation.

SACR also distinguishes among vendor positioning, technical documentation, product demonstrations, proofs of concept, customer references, and independently confirmed production outcomes. Roadmap functionality is not credited as current capability, and a demonstration establishes that a workflow can be shown rather than that it operates reliably at scale. Stronger evidence includes verification that the intended control remained in effect after deployment.

Sponsorship and Research Disclosure

This report reflects SACR's independent research and analysis of Cloud Security Intent Enforcement as of October 2026.

Aryon, Native, Act, and Wiz participated in sponsored research programs associated with this report. Sponsorship provided access to briefings, product demonstrations, and supporting materials. It did not influence SACR's conclusions, analysis, or editorial direction.

The vendor profiles illustrate different approaches to the control problems examined in this report. SACR acknowledges that this is a non-exhaustive sampling of vendors. There exist other specialist and incumbent solution providers who are relevant to CSIE. The vendor profiles are not product rankings or procurement recommendations. All opinions expressed are those of SACR.

Conclusion

Cloud teams already have plenty of tools telling them what is wrong. The harder part is managing what happens next across AWS, Azure, GCP, and the growing number of controls, policies, exceptions, and provider-specific requirements that come with them.

CSIE is ultimately about making that work easier. Instead of asking practitioners to translate the same security intent across different cloud providers, assess the likely impact of every change, and manage enforcement and exceptions by hand, these platforms can take on more of that operational burden.

The framework represents a coherent cloud-security control problem and emerging control layer, but its eventual product packaging remains unsettled. Specialists may continue to develop standalone platforms, while CNAPP, identity, policy-as-code, and cloud-provider platforms absorb selected capabilities.

When CSIE works as intended, practitioners should spend less time translating policies, chasing configuration changes, and managing provider differences, and more time deciding what the environment should actually allow. Fundamentally, CSIE shifts the operating burden from repeated remediation toward maintaining enforceable cloud intent.

Software Analyst®
Cyber Research

business personal

softwareanalyst.io

SACR®

Trusted research. Sharp insights. Real conversation.

CISO	VENDOR
SECURITY TEAMS	INVESTORS