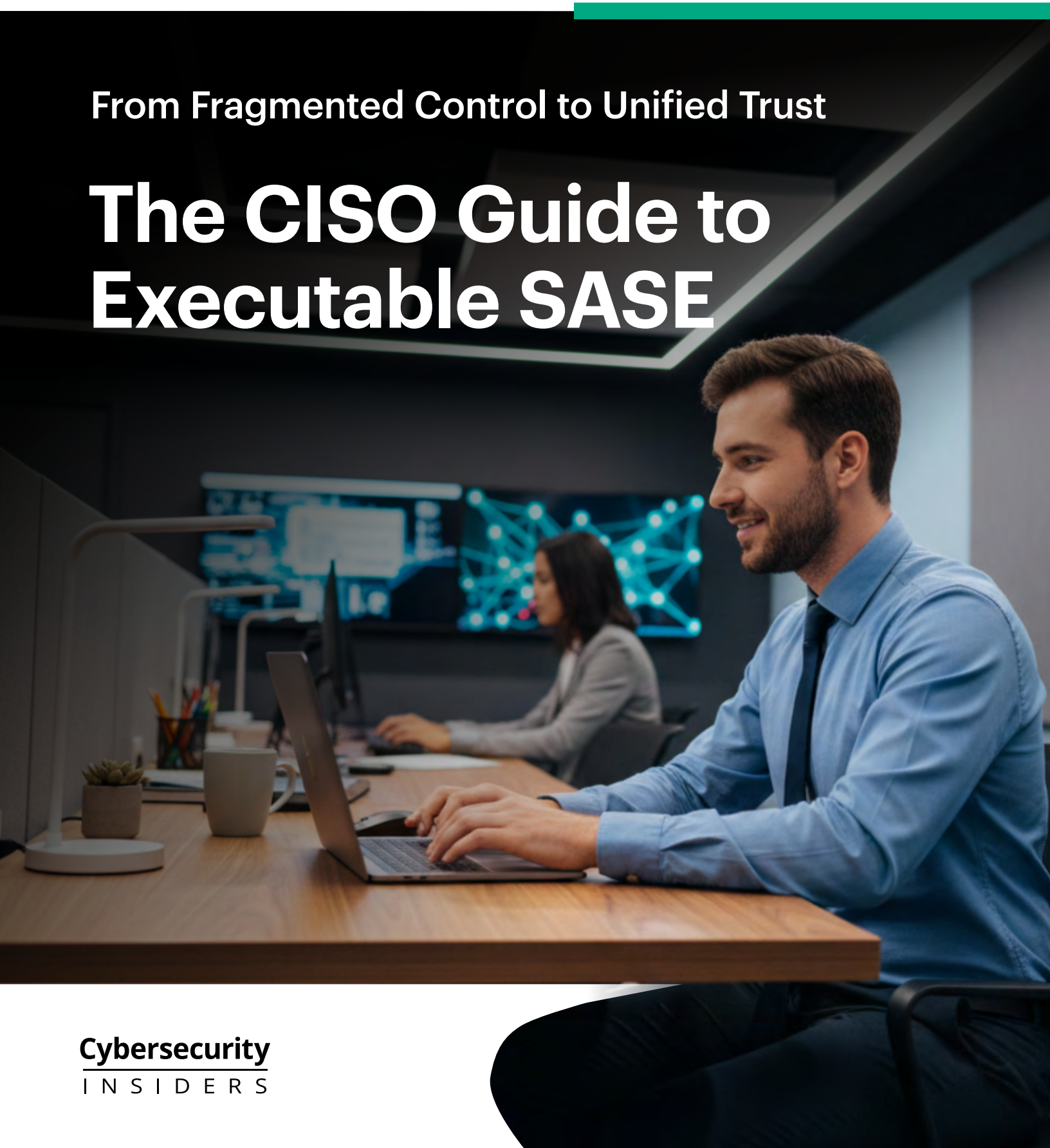
HPE

From Fragmented Control to Unified Trust

The CISO Guide to Executable SASE



Cybersecurity
INSIDERS

Executive Summary

Zero Trust has moved from aspiration to mandate across the enterprise, yet the gap between strategy and execution remains wide. While 82% of organizations view universal Zero Trust enforcement as essential, only 17% have fully implemented it (HPE/Cybersecurity Insiders, 2026 Zero Trust Report).

The gap is rarely about intent. Most security teams know what needs to happen. The challenge is translating that knowledge into consistent control across fragmented policy engines, disconnected telemetry, and operational environments that exceed what any single tool was designed to govern. That gap carries measurable cost: $4.88 million per breach on average, with detection still taking over 190 days (IBM, 2024).

Regulatory pressure is accelerating the urgency. NIS2 now holds management personally accountable for access control. DORA requires provable ICT controls subject to direct audit. SEC disclosure rules demand material incident reporting within four business days. The common thread: auditors expect enforcement demonstrated at runtime, with evidence, not policies described on paper.

Secure Access Service Edge (SASE) addresses this by converging networking and security into a unified, cloud-delivered architecture that enforces Zero Trust across branch, campus, remote, and cloud access paths. But the label alone is not enough. Many implementations modernize remote access or reduce tool sprawl while leaving campus, branch, and non-user environments under separate trust models — reorganizing complexity rather than resolving it. That persistent gap is why 63% of organizations now favor single-vendor or hybrid platform SASE (HPE/Cybersecurity Insiders, 2026 Zero Trust Report).

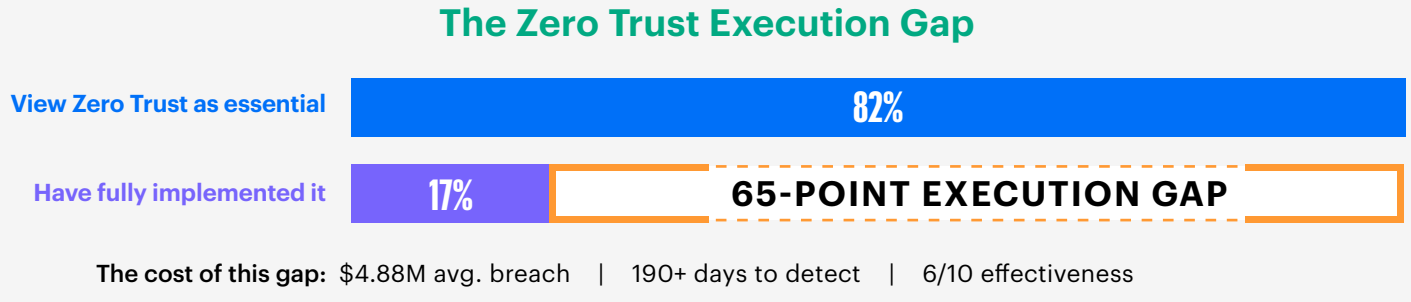
This guide lays out what separates executable SASE from architectural ambition, and how to evaluate whether a platform can actually close the gap. It provides a control maturity model, diagnostic questions to assess your current position, and evaluation criteria for vendor conversations, before examining how one architecture maps to those requirements in practice.

For CISOs under pressure to prove control to boards and regulators, executable SASE is how Zero Trust moves from policy to practice.

The Control Problem:
Why Zero Trust Fails at Scale

Zero Trust: Universal Strategy, Uneven Execution

Most enterprises have embraced the principles — verify explicitly, enforce least privilege, assess trust continuously. The challenge is rarely the framework. It is the gap between how policies are designed and how access is actually enforced across hybrid environments where users, devices, and applications move faster than the architectures built to secure them.



Source: HPE/Cybersecurity Insiders, 2026 Zero Trust Report; IBM Cost of Data Breach 2024

Dynamic Access Meets Static Infrastructure

The speed of that shift is outpacing the infrastructure meant to support it. Users work from everywhere, applications are SaaS-delivered, and data flows across multiple networks and clouds. Access increasingly involves automated systems and non-user devices that were never part of the original security model.

Much of the underlying infrastructure remains rooted in perimeter-centric control models. Traffic is still forced through centralized enforcement points designed for predictable, on-premises networks. A healthcare system backhauling branch clinic traffic to a regional data center for inspection adds latency that clinicians work around by finding shortcuts. That pattern repeats across industries: the architecture creates friction, users find bypasses, and security teams lose visibility at exactly the points where enforcement matters most.

The result is uneven control. Security teams enforce strong policies for remote users connecting through managed channels while losing consistency across branch networks, campus environments, and direct-to-cloud access paths. The same user, accessing the same application, faces different security controls depending on whether they are remote or on-premises. Zero Trust applies in principle. Enforcement remains situational.

That inconsistency traces back to how security architectures grew. As environments expanded, security toolsets expanded alongside them. The average Fortune 500 company now manages approximately 76 security tools, yet research suggests only a quarter deliver meaningful productive value (Ponemon Institute). Network security, access security, cloud security, and endpoint controls are frequently managed as separate systems, each with its own policy model, management interface, and telemetry.

The fragmentation is measurable: 78% of organizations manage secure-access policy across multiple systems. Only 17% operate from a unified platform (HPE/Cybersecurity Insiders, 2026 Zero Trust Report).

This fragmentation creates two persistent problems — inconsistent control and degraded response — each feeding the other.

Where Control Breaks Down

78% manage policy across multiple systems — **17%** operated from unified platform

Inconsistent control	Degraded response
Visibility gaps Networking and security teams lack shared context across users, devices, and data flows	**Slow containment** Detected threats stay uncontained while enforcement catches up across controls
Policy drift Access rules manually synced across systems; exceptions accumulate silently over time	**Audit friction** Evidence of access and enforcement assembled after the fact from multiple systems

Source: HPE/Cybersecurity Insiders, 2026 Zero Trust Report

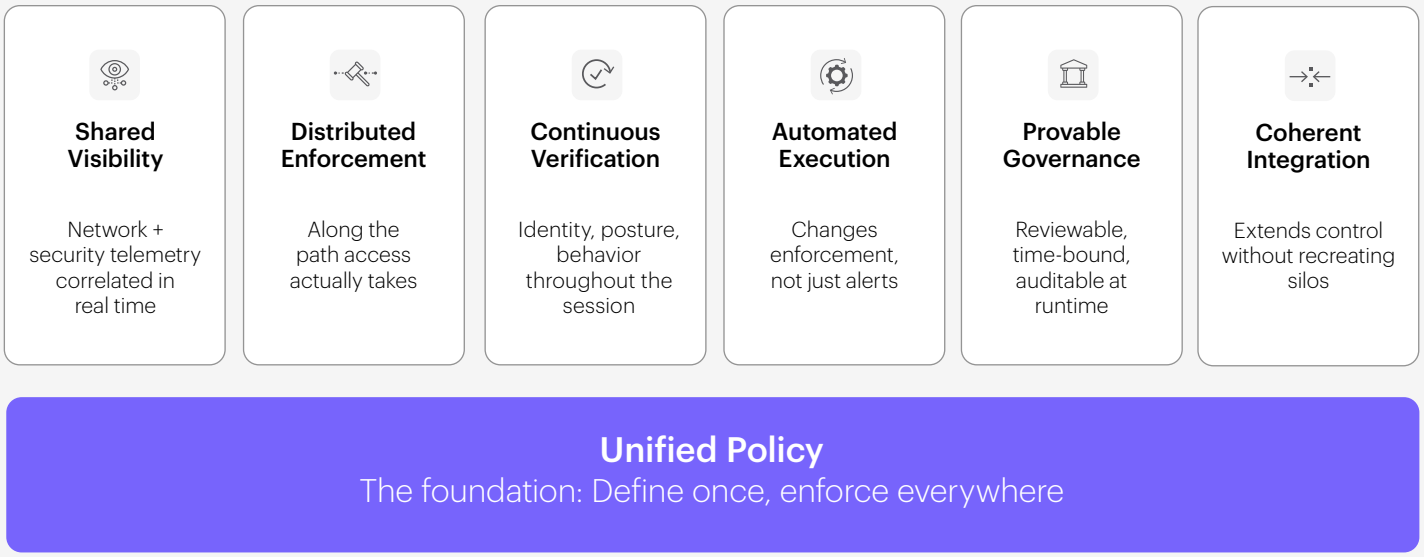
When risk changes, enforcement lags. When audits occur, evidence must be assembled from disconnected systems. SASE addresses this by converging networking and security into a unified, cloud-delivered architecture. But the label alone is not enough — many implementations reorganize complexity rather than resolve it. For CISOs, the challenge is no longer whether SASE matters, but how to identify architectures capable of executing Zero Trust consistently at scale.

Executable SASE:
The Design Principles That Matter Most

Zero Trust is the strategy — the security intent. SASE is the architecture, the means by which that intent gets delivered. SASE operationalizes Zero Trust only when policy translates into continuous enforcement where access actually occurs. Without native control of the network edge, enforcement stays uneven: strong for remote users, fragmented across campus, branch, and non-user environments like IoT.

Not every SASE platform meets that threshold. Some converge networking and security in name while maintaining separate policy engines, telemetry silos, and enforcement models underneath. For CISOs evaluating platforms, the distinction that matters is whether a SASE architecture can reliably turn policy intent into consistent control where access actually occurs. The following principles define what executable SASE must deliver in practice.

Design Principles for Executable SASE





Unified Policy

Intent must be expressed once and enforced everywhere — a single, coherent policy model governing access consistently across users, devices, applications, and data. That includes how data is accessed, shared, and moved across SaaS, cloud, and edge environments. Architectures relying on multiple policy engines introduce translation layers that become sources of drift, delay, and exception sprawl. A policy restricting access to financial systems should apply identically whether the user is remote, on campus, or connecting from a partner network. Security and networking must operate as a unified control plane from the start.



Shared Visibility

Consistent control depends on shared context. Networking and security telemetry must feed the same operational view — correlated in real time across users, devices, applications, and data flows. Today, most enterprise security teams piece together events across 3–5 separate consoles after an incident occurs, assembling context while the blast radius expands. Unified visibility keeps policy decisions and enforcement actions synchronized as conditions change. Fragmented visibility slows response and weakens governance.



Distributed Policy Enforcement

Policy is only effective when enforced along the paths where access actually occurs — across branch, campus, remote, and cloud environments, not funneled through centralized chokepoints that no longer reflect real usage. Excessive backhauling introduces latency, degrades user experience, and encourages the bypass behaviors that most severely undermine security. A manufacturing company routing all branch traffic to a central data center for inspection creates the same friction pattern as the healthcare example earlier: users find workarounds, and security teams lose visibility. Enforcement must operate close to users and applications, at throughput levels that avoid forcing tradeoffs between performance and protection.



Continuous, Context-Aware Verification

Trust decisions must adapt as conditions change. Identity, device posture, location, behavior, and risk signals all need continuous evaluation throughout a session — not just at login. When a device falls out of compliance mid-session, access should tighten immediately rather than persisting until the next authentication event. The goal is verification that preserves both security and usability, tightening access or prompting additional checks while keeping friction low for legitimate users.

Automated Execution

Manual coordination breaks down in hybrid, distributed environments. Automation must be embedded directly into the control plane to reduce operational friction and improve consistency — but the kind of automation matters. Isolating a compromised device, stepping down access privileges, triggering re-verification: these are enforcement actions that reinforce policy intent. Applied to unified control, automation increases resilience. Applied to fragmented systems, it amplifies instability. The test is whether automation can act on the control plane or merely accelerate alerts and ticket creation.

Provable Governance

Policies should be reviewable and traceable. Exceptions must be visible, time-bound, and accountable. Enforcement effectiveness must be demonstrable at runtime — from live operational data, not inferred from configuration screenshots. An auditor asking "show me who accessed this system last Tuesday and what policy governed that session" should get an answer in minutes, from a single source. Governance at this level transforms Zero Trust from a conceptual model into an operational and regulatory reality.

Coherent Integration

Most enterprises operate within complex ecosystems of identity providers, SOC platforms, and compliance systems. Open APIs and ecosystem connectivity should extend unified policy, shared telemetry, and consistent enforcement outward — not recreate silos or introduce parallel control planes. A SIEM integration that pulls correlated telemetry from the SASE control plane extends visibility; one that requires a separate data pipeline to function fragments it. The test is simple: does connecting a new tool extend the control plane, or fragment it?

These seven principles form the evaluation baseline for executable SASE. The question for most CISOs is not whether they agree with them, but how far their current architecture is from delivering them — and where to start closing the gap.

How CISOs Should Approach SASE Modernization

SASE modernization is a sequence of architectural and organizational choices that determine whether Zero Trust becomes operational reality or remains aspirational. Three diagnostic questions assess your starting point. Three decisions shape the path forward.

Most organizations can articulate their Zero Trust strategy clearly. Fewer can demonstrate that it operates consistently in practice. These three questions test that distinction — whether the design principles from the previous section are operational in your environment, or still aspirational.

Assess Your Starting Point

Policy Consistency

Can you demonstrate consistent policy enforcement across all access paths today: remote, branch, campus, and cloud?

Shared Visibility

Do networking and security teams operate from shared context, or do they rely on after-the-fact correlation?

Provable Control

Can you produce runtime evidence of who accessed what, under which policy, on demand — from a single source rather than assembled after the fact?

Organizations that answer "no" to all three face foundational challenges that shape every subsequent decision. Those with partial capabilities can target specific gaps. Either way, three architectural decisions determine whether the path forward strengthens control or introduces new risk.

Three Decisions That Shape the Path Forward

The diagnostic questions reveal where control is weakest. These three decisions determine how to address those gaps in the right sequence — starting with scope, then architecture, then pace of change.

DECISION ONE The Scope of Zero Trust Enforcement

A common early misstep is treating SASE primarily as a remote-access initiative. Replacing VPNs with modern access controls delivers immediate benefits but often results in a "remote-only" Zero Trust model. Remote users are tightly verified and constrained while on-premises users operate within flat, implicitly trusted networks. Risk is redistributed, not reduced. Flat trust zones persist, and lateral movement paths stay open.

CISOs should plan for universal Zero Trust scope by design, even if rollout is incremental. Early architectural choices must support consistent enforcement across remote users, branch offices, campus environments, and edge locations without introducing parallel policy engines or separate security models.

DECISION TWO Unified Versus Best-of-Breed Architecture

Consolidation is a control decision, not a procurement preference. Where fragmentation directly undermines enforcement consistency, response speed, or governance, unification should take precedence over component diversity. Multiple policy engines, telemetry silos, and manual coordination points slow response and increase drift between intent and enforcement.

The question is not whether to keep existing investments, but whether those investments operate under a shared control plane or require constant synchronization across independent systems. Organizations evaluating SASE platforms should count the number of independent policy authorities in the proposed architecture — each one is a potential source of drift.

DECISION THREE Sequencing Change Without Increasing Risk

Sequencing starts with control objectives, not connectivity upgrades. Attempting to unify policy, enforcement, visibility, and automation simultaneously overwhelms teams and increases operational fragility. Effective SASE modernization proceeds in deliberate stages, each strengthening control while preparing the foundation for the next.

In practice, this typically means prioritizing visibility before consolidation, and consolidation before automation. An organization might begin by establishing shared telemetry across its highest-risk access paths, then converge policy management and enforcement for those paths, and only then layer in automated response. Automation applied to fragmented systems amplifies instability rather than reducing it.

Measuring Control Maturity

The three decisions define scope, architecture, and sequencing. The maturity model below helps CISOs benchmark their current position against the design principles. Each level reflects how many of those principles are operational, and what advancing to the next level requires in practice.

SASE Control Maturity Model

Most enterprises operate between maturity Levels 1 and 2

LEVEL 1 — Fragmented

Multiple policy engines, manual coordination. Zero Trust started but enforced inconsistently.

Typical indicators:

- 3+ independent policy engines
- Audit evidence assembled manually
- Exceptions tracked informally

LEVEL 2 — Consolidating

Shared visibility emerging. Policy authority moving from many systems to few.

Typical indicators:

- Shared telemetry across primary paths
- Policy in 1–2 systems for most users
- Initial joint NOC/SOC operations

LEVEL 3 — Unified

Single policy model across all access paths. Enforcement consistent and auditable.

Typical indicators:

- One policy authority
- Runtime evidence on demand
- Network + security metrics reported jointly

LEVEL 4 — Adaptive

Automation reinforces unified control. Governance provable on demand.

Typical indicators:

- Automated containment for defined threats
- MTTC measurably reduced
- Evidence generated automatically

PRIORITY 1 — Stabilize
Close blind spots. Establish shared visibility first.

PRIORITY 2 — Simplify
Converge policy, telemetry, enforcement. Reduce control planes.

PRIORITY 3 — Automate
Embed adaptive response into unified control. Reinforce policy, not alerts.

Technical maturity advances only as far as the operating model supports it. Consistent control depends on networking and security teams operating from shared metrics — measuring performance and effectiveness together, not in silos. Incident response must operate from a single view without handoffs between NOC and SOC. And infrastructure and security roadmaps need to be developed in coordination from the start. Organizations that defer this alignment typically find that technical convergence stalls. Teams revert to familiar silos, and the benefits of unified architecture erode.

HPE

What CISOs Should Look for in a SASE Platform

The strategic requirements for executable SASE are clear: unified policy, consistent enforcement, shared visibility, adaptive control, and provable governance. The remaining challenge is determining whether a given platform can deliver these outcomes in practice.

An Architectural Distinction Worth Evaluating Early

Some SASE vendors begin from a security-only foundation, building SSE capabilities and then partnering with or acquiring SD-WAN. This can deliver strong cloud security, but enforcement at the campus, branch, and edge often depends on integration seams between separately designed systems. Architectures that build security enforcement natively into the networking edge avoid this seam entirely.

The following questions test each design principle directly — structured as evaluation criteria for vendor conversations and proof-of-concept exercises:

UNIFIED POLICY

Can Policy Be Defined Once and Enforced Everywhere?

Count the number of independent policy engines. If policies must be duplicated or reconciled between systems, fragmentation has been reintroduced. Test whether a single policy change propagates consistently across remote, campus, branch, and cloud access paths — or whether each requires a separate update.

SHARED VISIBILITY

Does Visibility Come from a Shared Foundation?

Ask to see one view correlating identity, device posture, network path, and policy enforcement for a single session. If this requires switching consoles, visibility is shared in name only.

DISTRIBUTED ENFORCEMENT

Does Enforcement Reach Every Access Path Without Forcing Tradeoffs?

Verify coverage across campus, branch, remote, cloud, and edge — including IoT and OT devices that cannot run agents. If non-user environments require a separate security model, the architecture has a blind spot. Confirm that control is applied at the edge, close to users and applications, at line rate. Architectures dependent on centralized backhauling introduce latency and encourage the workarounds that undermine enforcement.

CONTINUOUS VERIFICATION

Does Verification Continue Throughout the Session?

Ask the vendor to demonstrate what happens when a device falls out of compliance mid-session or a user's risk score changes after authentication. Does access tighten immediately, or does it persist until the next login? If the architecture only evaluates trust at the point of connection, it is checking credentials, not verifying trust.

AUTOMATED EXECUTION

Does Automation Act on the Control Plane?

Ask vendors to demonstrate automated response that changes enforcement: isolating devices, adjusting access, tightening policy. If automation only accelerates alerts and ticket creation, it adds operational load without reducing risk.

PROVABLE GOVERNANCE

Is Governance Provable at Runtime?

Ask how exceptions are created, tracked, and expired. If the answer is "manually" or "in a separate system," governance is bolted on. Enforcement evidence should be producible on demand from live operational data.

COHERENT INTEGRATION

Does Integration Extend the Control Plane or Fragment It?

Ask how the platform connects to existing identity providers, SIEM, EDR, and compliance systems. If integration requires a separate data pipeline, parallel policy engine, or standalone console, it recreates the silos the architecture is meant to eliminate. Open APIs should extend unified telemetry and enforcement outward, not require manual synchronization between systems.

Implementing Executable SASE:
The HPE Aruba Networking Approach

HPE Aruba Networking delivers those capabilities through two product families — EdgeConnect SD-WAN and cloud-delivered SSE — managed through a single control plane. This section maps the platform to the evaluation criteria above.

The HPE Aruba Networking Approach: A Unified Fabric

HPE Aruba Networking approaches SASE from the premise that the network itself is a primary enforcement surface, not a passive transport layer. This networking-first design executes Zero Trust where access occurs, rather than relying on downstream overlays to compensate for fragmented control. The architecture delivers executable SASE through two product families and a unified control plane.

HPE Aruba Networking: A Unified SASE Platform

EdgeConnect SD-WAN

- Secure WAN edge
- Built-in NGFW
- IoT/OT behavioral baselines
- Line-rate enforcement

Security Service Edge (SSE)

- ZTNA
- SWG
- CASB
- DLP
- Agent-based and agentless
- 500+ cloud edges

HPE Aruba Networking Central

Unified Management • Single Policy Authority • AI-Powered Observability • Open APIs

Remote Users • Branch Offices • Campus • IoT/OT • Cloud/SaaS

Combined Outcome: Universal ZTNA — identical enforcement for every user, device, and location

1 High-Performance Edge

(HPE Aruba Networking EdgeConnect SD-WAN)

Most SASE implementations require dedicated security appliances at each branch location or force all traffic through cloud inspection points — multiplying cost, adding latency, and leaving non-user devices (IoT, OT) outside the enforcement perimeter entirely. EdgeConnect SD-WAN eliminates this gap by converging security enforcement into the WAN edge, steering traffic based on application and path quality while enforcing policy where access originates.

> *EdgeConnect closes the gap most SASE implementations leave open, extending enforcement to non-user environments where agent-based security is impractical but risk is substantial.*

Key capabilities:

- **Converged security stack:** Integrates SWG, CASB, FWaaS, and DLP directly into the SD-WAN platform, eliminating the need for separate security appliances or third-party bolt-ons.
- **IoT and OT protection:** Extends Zero Trust enforcement to devices that cannot run agents (sensors, industrial controls, medical equipment).
- **AI-powered IoT security:** Uses AI-established behavioral baselines to detect anomalies and isolate compromised devices.
- **Line-rate enforcement:** Applies policy at the edge without backhauling traffic to centralized inspection points, preserving performance while maintaining control.
- **Natural language queries:** Enables frontline analysts to query the unified fabric directly, reducing the skill ceiling for day-to-day investigation.

2 SSE: Cloud-Delivered Security

(Universal ZTNA — Campus, Branch, and Remote)

Most ZTNA implementations protect remote users effectively but leave on-premises environments governed by flat, implicitly trusted networks — creating a two-tier enforcement model where the same user faces different security controls depending on whether they're in the office or at home. Universal ZTNA closes this gap by extending identical Zero Trust enforcement to every access path.

> *A policy restricting access to financial systems applies equally whether the user is remote or on campus. That consistency is what separates executable SASE from label-only implementations.*

Key capabilities:

- **All user types:** Remote workers, on-premises employees, contractors, partners, and third parties — verified consistently regardless of location.
- **All device types:** Managed endpoints, BYOD, IoT sensors, and OT systems, including enterprise edge devices that cannot run traditional agents.
- **All protocols:** VoIP, peer-to-peer, server-to-client, SSH, RDP, Git, and database connections — protocols that challenge most ZTNA implementations.
- **Local traffic stays local:** ZTNA Private Edge applies the same cloud-defined policies on-premises, keeping on-premises traffic on-premises, eliminating the inefficient "hairpin routing" and maintaining enforcement during connectivity disruptions.

3 Unified Control and Shared Visibility

(HPE Aruba Networking Central)

Most enterprise environments require security teams to correlate events across 3–5 separate consoles after an incident occurs, assembling context from disconnected telemetry sources while the blast radius expands. HPE Aruba Networking Central eliminates this delay by providing shared operational context from a single control plane across wired, wireless, WAN, and cloud environments.

> *HPE Aruba Networking Central gives networking and security teams a shared operational view, reducing the handoffs and blame-assignment patterns that slow incident response.*

Key capabilities:

- **Unified telemetry:** Networking performance signals and security events feed the same data foundation, enabling real-time correlation rather than post-incident log analysis. Enforcement evidence — who accessed what, under which policy, and when — is producible on demand from this shared foundation.
- **Single policy authority:** Access policies are defined once and enforced consistently across all network domains, eliminating the synchronization burden that fragments most enterprise environments.
- **Open integration:** Pre-built APIs connect to existing identity providers (Okta, Azure AD, Ping), EDR platforms, and SIEM systems — extending unified telemetry outward without reintroducing control silos.

Organizations moving from fragmented tools to a unified fabric report measurable gains across operational efficiency, enforcement consistency, and user experience. The market trajectory confirms this direction: **63% of organizations now favor single-vendor or hybrid platform SASE** over fragmented multi-vendor approaches (HPE/Cybersecurity Insiders, 2026 Zero Trust Report). The architecture supports stepwise adoption. Governance and auditability remain intact at every phase, making control effectiveness observable rather than assumed.

The design principles define what executable SASE must deliver. The evaluation questions test whether a platform can deliver it. What remains is the evidence — the operational outcomes and board-ready metrics that prove the investment is working.

Measuring SASE Success

OUTCOME	METRIC	PLATFORM ENABLER
Consistent security policy enforcement	Reduction in independent policy definitions and enforcement points	Single policy authority across all network domains
Shared visibility across networking and security	Number of consoles required to investigate an incident; time to correlate identity, device, network, and policy data for a single session	Unified telemetry across wired, wireless, WAN, and cloud
Enforcement across all access paths without tradeoffs	Percentage of access paths (campus, branch, remote, cloud, IoT/OT) governed by consistent policy at line rate; reduction in VPN-related and access-related support tickets	Edge enforcement + Universal ZTNA coverage across all user and device types
Stronger identity- and posture-aware access control	Percentage of application access governed by identity, device posture, and context checks; MFA enforcement rate	Universal ZTNA per-session identity- and context-based verification
Automated threat containment	Percentage of defined threats contained without manual intervention; reduction in mean time to contain	AIOps-driven automated response acting on the unified control plane
Provable governance and audit readiness	Time to produce enforcement evidence on demand; percentage of exceptions tracked and time-bound	Unified telemetry foundation and policy traceability
Reduced operational complexity and cost	Number of tools retired, consoles reduced, and FTE hours reclaimed	Unified management, telemetry, and automation

By tying architectural decisions to observable outcomes and measurable metrics, SASE shifts from a technology deployment to a measurable control program — providing the evidence boards expect when evaluating risk posture and ongoing investment.

From Strategy to Day-to-Day Control

The shift to executable SASE reflects a broader evolution in the CISO role: from managing tools and exceptions to designing systems of control that scale with the business.

This guide defined what executable SASE requires — seven design principles, a maturity model for benchmarking progress, and evaluation criteria for distinguishing platforms that deliver consistent control from those that reorganize complexity. The gap between Zero Trust as policy and Zero Trust as practice defines which organizations can demonstrate control under pressure. Closing that gap demands an architecture where enforcement is continuous, visibility is shared, and governance is provable — not just designed, but operational.

For CISOs navigating expanding attack surfaces, AI-driven threats, and intensifying regulatory scrutiny, executable SASE is how Zero Trust moves from policy to practice.

NEXT STEP: CISOs looking to benchmark their current posture can begin with an architectural assessment to inform sequencing and investment decisions.

Explore HPE Aruba Networking SASE — including interactive workshops and hands-on evaluation options.

www.hpe.com/us/en/networking/sase