BUYER'S GUIDE

The Buyer's Guide to Unified API Platforms

What to Evaluate Before You Build or Buy Integrations

The Hidden Cost of Building Integrations

Every integration starts with a reasonable assumption: we have engineers, we have access to the vendor's API documentation, and we can ship this in a sprint. Synqly's State of Cybersecurity Integrations 2026 report — surveying senior leaders across product management, engineering, alliances, and executive leadership — reveals something different.

Integrations routinely take two to three times longer to build than planned. Sixty-nine percent of product leaders cite sales or prospects as the primary driver behind building integrations, while 54% also cite partner ecosystem programs as a strategic driver. Most integrations are reactive, not strategic.

2–3×	69%	54%
Longer to build than planned	Built in reaction to sales pressure	Cite partner ecosystem as driver

Why Internal Integration Development Is a Strategic Liability

The case for building in-house is intuitive. Engineering teams understand the product, they control the codebase, and a bespoke integration always seems faster than evaluating a third-party platform. The problem is not the first integration. It is what the second, fifth, and fifteenth cost — in engineering time, maintenance overhead, and the organizational debt that accumulates when integration work displaces product work.

The Setup Phase: Where Time Goes Before Development Begins

Research participants consistently identified three categories of pre-development work that extend integration timelines well beyond what engineers initially scope:

- **Establishing the partnership.**
  Third-party integrations require access to the target product. Best practice is forming a Technology Alliance Program agreement — which typically takes three to eight weeks.
- **Gaining access to an NFR environment.**
  Development teams cannot test against customer production data. Obtaining a Not-for-Resale license requires a formal request, legal review, and provisioning time.
- **Technical preparation and use case development.**
  With an environment available, teams review API documentation, set up integration scaffolding, and define the specific use cases the integration must support.

> Bypassing the setup phase by building against a customer's live environment creates zero regression testing exposure, version fragility, bespoke bias that does not generalize, and direct GDPR/SOC 2 compliance risk. It is not faster — it defers the failure.

Maintenance, Normalization, and the Measurement Gap

The Long Tail: Maintenance as a Variable, Ongoing Cost

If the setup phase is the underestimated upfront cost, maintenance is the cost that never closes. Most respondents reported that ongoing maintenance consumes more engineering time than the initial build — and the variance is high. The dominant driver is API instability: third-party APIs change, vendors push updates without changelogs, and integrations break in production.

Because many organizations lack visibility into which users are tied to specific integration configurations, failures are discovered by customers before they are discovered by engineering. Respondents unanimously agreed that finding a better way to handle API changes was essential to managing long-tail costs.

70%	9 of 13	2–4×
Named API instability as top driver	Had no integration health visibility	Maintenance vs. initial build cost

The Normalization Problem: Why Point-to-Point Does Not Scale

Seventy percent of product leaders named the lack of an abstraction layer as a primary barrier to scaling integrations. Every new vendor introduces new schemas, field names, authentication patterns, and rate limit behaviors. Without a normalization layer, each integration is a one-off. As the integration portfolio grows, the surface area of bespoke code expands — and so does the engineering burden required to keep it running.

The Measurement Gap: Flying Blind on Integration Value

Nine of thirteen research participants do not measure the business value or ROI of their integrations. Without measurement, integrations check a box for one customer, contribute minimally to the broader product roadmap, and are deprioritized relative to features with clearer revenue attribution. Teams cannot prioritize, cannot justify platform investment, and cannot course-correct.

Three Foundational Capabilities Every Unified API Platform Must Have

Not all unified API platforms are equivalent. Some are thin wrappers around public APIs — convenient until a vendor changes a field name or deprecates an endpoint. These three capabilities are non-negotiable baselines. Any platform that cannot demonstrate all three at depth is not a platform; it is a shortcut.

Foundational and Advanced Capabilities

1 Configuration Management

Centralized, programmatic control over every integration in the portfolio. Look for: tenant-scoped configuration, API-driven credential rotation, and environment-level overrides without redeployment. This is the difference between a portfolio that scales with your customer base and one that requires manual engineering engagement per customer per change.

2 Bi-directional Data Access

Read-only integrations are a starting point, not a destination. As integrations mature toward orchestrated response, write access is required. Look for: write operations that are governed, auditable, and role-scoped through a consistent API surface.

3 Standardized Schemas

Seventy percent of security product leaders named the lack of standardized data formats as a primary barrier to scaling. Look for: native OCSF alignment as a baseline — not a bolt-on normalization layer that transfers the normalization burden back to your engineering team.

Advanced Capabilities: Credentials, OCSF, Query Language, Deployment, MCP

Beyond the three foundational capabilities, five additional capabilities define the platforms that deliver durable value in production environments.

Credentials Management

API keys, OAuth tokens, and service account credentials across dozens of systems require centralized rotation, scoping, and audit trails. A platform that does not manage this transfers the operational risk — and the security exposure — to the customer.

OCSF Standards Support

Native OCSF alignment enables participation in regulated procurement and FedRAMP-compatible environments, and reduces proprietary lock-in. A platform built on open standards can be audited, extended, and migrated without bespoke reengineering.

A Native Query Language

Integration data is only useful if it can be queried in ways that enable your product's use cases without building and maintaining a custom pipeline on top of the integration layer.

Flexible Deployment

Cloud, on-premises via a bridge architecture, and hybrid models are all required to serve enterprise security customers. Air-gapped and embedded deployment support closes deals in regulated verticals.

MCP Support

The Model Context Protocol is an emerging standard for connecting AI agents and LLMs to external data. Native MCP support allows AI agents to access the full integration surface without custom per-vendor tooling.

The Operational Layer and Evaluation Checklist

The Operational Layer Most Vendors Ignore

When engineering teams evaluate integration platforms, the technical capabilities get the most attention. What gets less scrutiny is the operational layer — the work required before any code is written and the monitoring required after any code ships.

Vendor Relationship Management

Technology Alliance Program agreements, Collaborative Support Agreements, and joint go-to-market commitments all precede technical access. A platform that maintains these relationships on behalf of its customers eliminates this overhead from the engineering team's scope. Ask: do you maintain active TAP agreements with the vendors in your connector ecosystem?

NFR License Access and Test Environments

Not-for-Resale licenses provide sandboxed pre-production environments. Obtaining them through normal channels takes weeks. A platform that manages NFR environments across its ecosystem allows engineering teams to begin development on day one. Ask: how long does it take to provision a test environment for a new integration?

Break and Change Monitoring

API instability is the single largest hidden cost of in-house integrations. A platform that actively monitors connected vendor APIs, tests against the live integration, detects schema drift, and alerts before customers encounter failures is the core operational promise of the category. Ask: how does your platform notify customers when a connected vendor API changes?

The Integration Evaluation Checklist

FOUNDATIONAL CAPABILITIES

- [ ] Bi-directional data access through a single normalized API surface?
- [ ] OCSF alignment native to the data model — not a bolt-on layer?
- [ ] Tenant-scoped configuration management without redeployment?

OPERATIONAL LAYER

- [ ] Active TAP agreements with vendors in the connector ecosystem?
- [ ] NFR test environment provisioning timeline for a new integration?
- [ ] Break and change monitoring with pre-customer alerting?

ADVANCED & DEPLOYMENT

- [ ] Credentials management built into the integration layer?
- [ ] On-premises, hybrid, and air-gapped deployment support?
- [ ] MCP support for AI agent access to the integration surface?

COMMERCIAL

- [ ] Cost model and how it scales with customer and connector growth?
- [ ] Uptime SLA and remediation process?
- [ ] Onboarding timeline from contract to first production integration?

Why Synqly

Synqly is The Connected Platform for Security, IT, and AI — built from the ground up for OEM deployment and enterprise environments, with a production-grade ecosystem of security and IT connectors. For security and IT software vendors evaluating how to deliver the integrations their customers require without indefinitely scaling an integration engineering team, Synqly provides the integration infrastructure.

The Largest Security and IT Connector Ecosystem

Synqly supports the broadest ecosystem of security and IT connectors available in a unified API platform, spanning application security, asset management, cloud security, endpoint (EDR), identity, SIEM, ticketing, vulnerability management, and more. Each connector is backed by active testing, relationship-based vendor access, and continuous break monitoring — not solely public APIs.

Integrate Once. Access the Category.

Synqly's fundamental architectural promise: integrate once into a category and gain access to every provider in that category, including providers added after the initial integration. This is the mechanism by which the build-once-maintain-forever model becomes possible. Integrating into SIEM means every SIEM connector, not just one.

Relationship-Backed Integrations — NDAs and NFR Licenses Included

Synqly manages the vendor relationships, Technology Alliance Program agreements, non-disclosure agreements, and NFR license access that would otherwise fall to the engineering team. The pre-development overhead that extends timelines by a full quarter is already handled.

Deployment That Meets Customers Where They Are

Synqly supports on-premises, cloud, and hybrid deployment environments. The Bridge architecture enables connectivity to systems behind firewalls and in air-gapped environments. The embedded deployment model supports FedRAMP-compatible environments for government and highly regulated enterprise customers.

Security, Compliance, and Reliability

Synqly is SOC 2 certified. The platform aligns with OCSF for data normalization and OSCAL for compliance data exchange. Synqly maintains a 99.9% SLA for platform uptime — a commitment that reflects the production criticality of the integration infrastructure it supports.

Built for What Integrations Are Becoming

Synqly's support for the Model Context Protocol (MCP) enables AI agents and LLMs to access the largest security and IT connector ecosystem via a unified, normalized API surface. Built around open industry standards — OCSF, OSCAL, and MCP — the platform grows with the industry rather than against it.

synqly™ | The Connected Platform for Security, IT, and AI

Synqly is The Connected Platform for Security, IT, and AI. Built from the ground up for OEM and enterprises, Synqly's patent-pending data orchestration and normalization layer streamlines product integrations, leveraging a unified API. Synqly delivers the secure integrations that customers demand, continuing to set the standard for scalable, secure, and efficient integrations across the industry.

Based on findings from The State of Cybersecurity Integrations 2026, authored by Doug Cahill. Commissioned by Synqly, 2026.