CLOUDSOFT

WHITEPAPER

Connectors as a Product to Accelerate Adoption: Defining the Future of Software & Cloud Coverage

Josh Krammes, Head of Strategy

cloudsoft.io

Connectors as a Product to Accelerate Adoption: Defining the Future of Software & Cloud Coverage

EXECUTIVE SUMMARY

Enterprise estates are complex. Cloud-native services sit alongside legacy systems, bespoke workflows and organisation-specific dependencies. Historically customer IT teams were left to integrate new ISV software. That is changing. Buyers lack the time, budget and capacity to wire in software themselves and now expect enterprise-ready, out-of-the-box delivery. Documentation exists, but the products that win remove manual integration and feel plug-and-play from day one. According to G2, more than 80% of enterprises now rank ease of integration ahead of cost when evaluating new solutions or incumbents.

The response to this challenge in this paper is a Push and Pull methodology delivered as connectors treated like products, not glue. Push shows up where users already work so action happens in context. Pull ensures your product arrives with the data, coverage and schema alignment that allow value to be realised day one. Across both modes the same discipline applies: abstract transport, align schemas, enforce strong authentication and authorisation, and expose operational signals so teams can run integrations with confidence.

This approach reduces time to value and lowers maintenance overhead while improving coverage across multi-cloud and third-party estates. It also aligns with how software is bought and deployed today, where pre-integrated, marketplace-ready offerings set expectations for speed, compliance and interoperability. In production, the impact is material. For example, a hyperscaler connector built and maintained on this approach has run for seven years and eliminated 91% of manual effort for enterprise customers.

The rest of this paper sets out the model and the engineering practice behind it, with anonymised case studies that show it in production. The aim is straightforward: help product and platform leaders deliver integrations that meet users where they work, arrive with useful data, and stay in sync at scale.

SECTION 1

Introduction

Connectors are the hidden plumbing of modern cloud operations. From ITSM ticketing integrations to security feeds and IaC automation, nearly every meaningful cloud transaction today flows through a connector. Yet despite their ubiquity, connectors are often built as one-offs by dev teams under deadline pressure, poorly documented or hard to maintain, slow to certify on cloud marketplaces, and lacking in observability, upgrade paths, and SLAs.

In a landscape that values speed, compliance, and interoperability these limitations are not just inconvenient, they're unsustainable. Enter Connectors-as-a-Product (CaaP): a model for how integration can work in the modern cloud.

Connectors-as-a-product (CaaP) posits that the right integrations and product coverage at scale, is a competitive advantage that can help you win more customers, drive adoption and displace competitor products quicker.

Why Now: Market Timing and Forces

Several converging trends make this the moment for CaaP to emerge as a category. Cloud marketplaces are exploding. By 2028, enterprise software sales through cloud marketplaces will reach $85 billion USD (Canalys, AWS). Microsoft's commercial marketplace alone transacted over $5 billion in 2023 and was growing 2x year-over-year at that time. Marketplace growth is driven by the increasing demand for pre-integrated, enterprise-grade software.

ISVs are under pressure to integrate quickly with AWS, Azure, GCP, and key platforms like ServiceNow, Dynatrace, and Splunk. However, legacy systems lag behind. Many enterprises still rely on outdated tooling not built for cloud-native environments, making integration an ongoing pain point. Engineering leaders are overwhelmed. In GitHub discussions and across dev forums, there's widespread frustration with the manual burden of integration, especially when certification or security compliance is required.

Worldwide public cloud services were projected to exceed $805 billion in 2024, with cloud infrastructure spend alone reaching $321 billion. Gartner forecasts cloud-related end-user spending will top $723 billion in 2025 (CIO Dive). Canalys reports that cloud channel sales, partner ecosystems, and integration revenue are outpacing IaaS growth.

These conditions make it clear: connectors are not just tactical glue, they are strategic assets. And they need to be treated with the same rigor as production software, especially when deployed across cloud marketplaces, partner ecosystems, or regulated environments.

Global teams of all sizes are now looking to the ISV community to build connector products to remove this friction in their technology stack. Without having these integrations and pre-built connectors customers struggle to adopt, engineering teams lose valuable time and deals are ultimately lost or delayed.

SECTION 2

Connector Fundamentals

A connector is a protocol-abstracting, schema-aligning integration that enables systems to securely and reliably exchange data, synchronize processes, and automate workflows across different environments. The best connectors do far more than pass data; they define, translate, and validate the structure (schema) of the information they handle, eliminating mapping errors and inconsistencies between source and target systems.

Robust connectors are secure by design, supporting authentication, mutual Transport Layer Security (mTLS), auditability, and granular role-based access controls. They also deliver operational observability through logging, health checks, and real-time error reporting. Maintenance is easier when connectors are certified for cloud marketplaces and can be rapidly patched to address dependencies and compliance changes.

Modern connectors mediate data context, abstract away transport protocols, support both push (event-driven) and pull (polled) modes, and can be extended to new use cases as platforms evolve. As businesses rely on multi-cloud, hybrid, and SaaS application stacks, these traits determine whether integration efforts scale or stall.

2.1 Push vs Pull

Adoption breaks in two moments: when users must leave their tools to act or when your product arrives empty without the data it depends on. Push addresses that gap by delivering your value directly into the environments where work already happens. Pull answers the second by ingesting and harmonising the events, resources and telemetry your product depends on.

2.1.1 Push - Meet Users Where They Work

When customers expect to access your product's capabilities from environments they already use they won't always come to a new console just to get value from a new product. Push makes your product usable inside those third-party experiences so action happens in-context, with status reflected back to keep sources of truth aligned.

Implementation focus (Push)

Optimise for the host UX: create/update the right records with the necessary context so people can act without switching surfaces.

Keep the loop closed: synchronise state back to your product to preserve ownership and resolution accuracy.

Make it operable: instrument the connector with health, logging and error visibility; enforce IAM and mTLS/RBAC as first-class controls.

2.1.2 Pull - Earn the Coverage Your Product Needs

Coverage is the difference between a demo and dependable value. Pull brings in the resource types, events and signals that make your product useful immediately. Many analytics and observability cases run effectively as pure Pull. Pull approaches can align schemas between products so customers get meaningful insights from one product where they need them in another product. Pull approaches can listen for triggers from code hosts or build systems to keep state in another product fresh.

Implementation focus (Pull)

Ingest at scale: onboard the resource types your customers expect and normalise data for consistent downstream use.

Wire the triggers: accept events from external systems (e.g., version control or build services) so your product reacts to real work, not polls alone.

Run unattended: design for observability, security and predictable performance; treat ingestion paths as product, not scripts.

2.1.3 Where they meet

Push and Pull are distinct problems, but often shipped together. Projecting your experience into another tool usually requires a small amount of Pull to reconcile status, while many scenarios remain cleanly Pull-only. The architecture should acknowledge both without conflating them.

2.1.4 One discipline across both

Regardless of direction, modern connectors do the same hard things well: abstract transport, align schemas, enforce strong authentication and authorisation, and expose operational signals so teams can run them with confidence. We see that the most successful teams accelerate adoption by treating connectors as products (not just glue) making integrations provide consistent UX across platforms, while being reliable, observable and ready to scale across end customers.

2.2 Why Connectors Need a Product Model

Enterprise connectors today are often fragmented, highly manual, slow and painful to maintain, and lack rigorous monitoring and support. This fragmentation impedes innovation and creates operational risk. By adopting a Connectors-as-a-Product (CaaP) model, teams centralize the design, certification, deployment, monitoring, and ongoing management of connectors. This enables faster integration cycles, proactive compliance, consistent updates, and a dramatic reduction in operational firefighting.

CaaP brings a product discipline to integration engineering: connectors become managed, testable, observable, and signed artifacts rather than brittle, bespoke code. It also supports security controls, role-based governance, marketplace certifications, and SLAs, offering the visibility and reliability modern cloud-native architectures demand (Source).

SECTION 3

Case Studies

Connectors-as-a-Product goes beyond basic API wiring. The model is established, and the results bear it out. Users stay in the tools they already use, your product arrives with useful data, and adoption accelerates. The following anonymised case studies show this in production across a global hyperscaler, an observability ISV and a configuration service. Different contexts with the same outcome of lower friction for users and faster adoption.

3.1 Global hyperscaler - meeting users where they work

A major cloud provider saw teams revert to familiar tooling because provisioning in the provider console felt like extra work. We built and now maintain a connector that standardises and simplifies connections to a leading ITSM platform used by thousands of large enterprises. In production for seven years, it eliminated 91% of manual effort for enterprise customers. Operators can provision, manage and operate cloud resources natively in the ITSM tool, with a light status sync keeping sources of truth aligned. The result: adoption driven by in-tool action, not console training.

3.2 Observability ISV - turning volume into value

Trials were healthy, data flowed, but conversions lagged. The blocker wasn't in adopting new software; it was the manual lift to configure logs, metrics and alerts across cloud, host and Kubernetes estates. We shipped blueprints in that pre-define high-value signals, default dashboards and alert policies. Freshness targets were explicit and visible; volume spikes were handled through sensible filtering, sampling and back-pressure rather than dropped data. The out-of-the-box experience became decisive: customers moved from raw telemetry to actionable views, and conversion to paid increased materially.

3.3 Configuration service - coverage at multi-cloud scale

Customers valued the power in assessment and audit, but implementation and rules authoring landed on them - and single-cloud coverage no longer matched reality. We delivered coverage at scale: connectors across hyperscalers and major third-party applications, aligned to a canonical rules model. Ingest paths normalised schemas, made recency visible, and absorbed bursts with incremental fetch and queueing. Customers now bring whichever environments they have and assess configurations immediately, with consistent results and faster onboarding across estates.

Pattern across all three: show up where users already work; arrive with the data and defaults that make day one useful; instrument the path so freshness and scale don't become tomorrow's problem.

3.4. AI and the Role of Connectors

The value of AI in the enterprise increasingly depends on access to high-quality, timely, and context-rich data from a wide variety of systems. Connectors are foundational enablers in this domain. They not only gather and move information, but support AI-powered mapping, transformation, and augmentation of data flows between applications at scale. Without full coverage or the right integrations in place, there's a risk services and products get left behind for more AI native connected products.

Next-generation CaaP solutions offer support for AI-driven schema mapping, predictive field matching, and intelligent anomaly detection within integration pipelines. This enables rapid onboarding of new data sources, faster rollout of automations, and real-time insights, all without demanding constant engineering intervention for each new API or data shape.

3.5 Marketplace-Ready Integrations

To maximize reach and adoption, connectors must meet the evolving standards of major cloud and SaaS marketplaces. This goes beyond security and reliability—it includes documentation, certification, compliance evidence, and native support for billing and entitlement models.

A CaaP approach operationalizes this process, streamlining the packaging, validation, and submission of connectors for AWS, Azure, Google Cloud, and enterprise SaaS marketplaces. Continuous integration pipelines automate certification checks, regression tests, and publishing, while ongoing observability ensures connectors remain healthy post-listing, helping solutions stay viable and competitive in a rapidly-changing ecosystem.

SECTION 4

Engineering Challenges in Cloud-Native Connectors

Implementing Connectors-as-a-Product is not trivial. Engineering teams are focussed on delivering new feature development and defined product roadmaps. Engineering leaders consistently cite the following challenges:

- **Distributed Complexity:** Cloud-native architectures are inherently complex and distributed, with countless APIs, microservices, and functions across hybrid environments. Monitoring connector interactions across these boundaries is difficult without centralized tooling. (Source)
- **Configuration and Data Risk:** Misconfigured SSL/TLS or retry settings can silently degrade performance or expose vulnerabilities. Stateless environments like serverless introduce additional risks around data persistence and message loss.
- **Compliance and Security:** Securing service-to-service communication with mutual TLS (mTLS), maintaining audit logs, enforcing RBAC, and passing marketplace certification reviews all require significant effort. These burdens multiply across customers without reusable frameworks.
- **Performance and Scalability:** Latency, throughput, and uptime are all impacted by connector design. Legacy scripts don't scale. Horizontal scalability and intelligent caching are required to meet modern SLAs.
- **Vendor Lock-in:** Connector code hardwired to specific platforms increases the risk of vendor lock-in and makes re-platforming difficult and expensive.
- **Maintenance Overhead:** Even after deployment, custom connectors are often brittle. Small API changes can break them. Teams are left firefighting outages caused by undetected regressions.

SECTION 5

Methodology: what to get right up front

The methodology that follows distils the upfront decisions we make on every engagement so integrations are consistent, observable, secure and maintainable at scale.

This methodology applies across Push and Pull. Set the crucial upfront calls - manage data volume, align schemas and field mappings - and build in security and operability, and you get integrations that scale without becoming brittle. The checklist that follows turns those engineering challenges into clear decisions.

Area	Upfront focus	Outcome
User context & UX	Setup UX (static mapping, optional AI); data freshness targets; consistency across sources/ targets.	In-context use; fast onboarding.
Security & connectivity	Least-privilege access; mTLS/RBAC; safe retries; audited changes; reversible configs.	Secure by default; certifiable.
Data schema & semantics	Canonical model; explicit 1:1 mappings; rules for “near-matches”; versioned contracts.	Clean joins; predictable behaviour.
Data flow & scale	Volume and query patterns; incremental vs batch; state ownership; idempotency/back-pressure.	Meets SLOs under load; resilience.
Business concerns	Portability to avoid lock-in; reusable scaffolding; observability; low-touch upgrades.	Lower TCO; fewer breakages.

SECTION 6

Engineering Challenges in Cloud-Native Connectors

Connectors-as-a-Product offers significant, quantifiable benefits over traditional, custom-built integrations. The following outline highlights how CaaP delivers value across technical, operational, and business dimensions:

- **Faster Time-to-Value and Reduced Development Overhead:** with a structured connector factory, new integrations can be built and certified in days, enabling faster rollouts and strategic agility.
- **Meet your customers where they are:** It's no longer enough to provide the tools for your customer to integrate your product, you need to be ready to plug-and-play. The right connectors allow you to do this.
- **Adoption Accelerant:** the quicker your customer can consume your offering, the quicker they can see the value and scale adoption.
- **Marketplace Consumption:** with the right marketplace integrations across your product, you remove friction not only in the buying process but in the deployment and implementation of your product in your customers accounts.

Connectors-as-a-Product delivers a wide range of benefits to engineering teams, product leaders, and within software & cloud service businesses. It accelerates adoption by using reusable factory patterns that reduce time-to-market. It improves cost efficiency by reducing the sales cycle and the sales to engineering demand for one off integrations.

CaaP is an approach that will allow software companies and cloud service providers to offer a plug-and-play solution to customers that is quick to implement and allows the customer to bring whatever cloud or 3rd party tooling they want to your solution.

Cloudsoft is the engine behind the integrations

Cloudsoft builds enterprise-ready, out-of-the-box software that is yours to own. We embed our engineers into your workflows to deliver robust integrations and tailored solutions that accelerate time to value. From onboarding through to adoption at scale, your product works in enterprise environments and across the ecosystems your customers rely on.

- ✓ Trusted by the world leaders in Cloud, Security, Observability to build technology that drives adoptions and reduces friction in new customer acquisitions.
- ✓ A proven Connector Factory. Evidence backed experience in bringing CaaP to the tech sector
- ✓ Deep Domain expertise. We have spent 14 years building technology that is used by 1000's of enterprises every day.

Connectors are not just pipes. They're the connective tissue of the cloud economy. As the cloud-native world grows more composable, secure, and marketplace-driven, a new model is needed. For ISV's, the message is clear: with deep and wide coverage across your customers' technical estates, you can accelerate adoption and consumption of your product, while offering a more unified customer experience.

Cloudsoft invites ISVs, cloud providers, and enterprises to adopt Connectors-as-a-Product as the new integration standard.

WANT TO DEFINE THE FUTURE WITH US?

GET IN TOUCH

CLOUDSOFT

cloudsoft.io