

Author: Alexei Balaganski

 **kuppingercole**
ANALYSTS

Buyer's Compass

API Security

AIRLOCK[®]

Chapter 1	The Challenge
Chapter 2	The Solution
Chapter 3	Top Use Cases
Chapter 4	Top Functional Selection Criteria
Chapter 5	Vendor Spotlight
Chapter 6	Considerations

The Challenge

Application Programming Interfaces (APIs) have become the primary access layer of the digital enterprise. They expose business logic, broker access to sensitive data, connect partners and ecosystems, and increasingly serve as the connective tissue for AI-enabled applications and autonomous agents. As a result, API security is no longer a narrow technical discipline but an architectural concern that sits at the intersection of application security, identity, and access control across cloud, on-premises, and hybrid environments.

The main challenge here is not just API sprawl, but fragmented enforcement. APIs are deployed behind different gateways, embedded in Kubernetes clusters, exposed through legacy web infrastructure, or consumed indirectly by mobile and AI-driven applications. Visibility is often incomplete, ownership unclear, and access decisions enforced inconsistently. Traditional perimeter controls offer little protection against misuse of legitimate interfaces or business logic abuse. What many businesses lack is a consistent way to secure the access to applications and APIs wherever they run.

AI adoption further sharpens this problem. Agentic systems chain API calls, act asynchronously, and operate with delegated authority. The primary risk is no longer an obviously malicious request, but a valid action executed in the wrong context. Addressing this requires strong authentication, fine-grained and continuous authorization, and the ability to enforce step-up controls or human approval for sensitive operations.

Enterprises require unified solutions that align with Zero Trust principles, delivering consistent policy enforcement, in-depth visibility, and compliance across the entire API lifecycle and across all environments. They must be able to address the following challenges:

Explosion of API Volume and Complexity

The average enterprise now operates hundreds or even thousands of APIs, with a growing mix of protocols and interaction models such as REST, GraphQL, gRPC, event streaming, and increasingly AI-oriented interfaces like the Model Context Protocol (MCP). This rapid growth, driven by microservices, cloud-native architectures, and AI integration, has outpaced many organizations' ability to track and manage their APIs. They simply evolve faster than governance models can keep up. Documentation lags behind implementation, ownership becomes unclear, and visibility into what exists in production quickly erodes.

Shadow APIs emerge that bypass security review and approval processes, while zombie APIs remain exposed long after their intended use has ended. Both create unnecessary attack surface and compliance risk. The problem is compounded in hybrid and multi-cloud environments, where APIs may be deployed across different gateways, service meshes, and clusters with inconsistent controls. This level of fragmentation not only in-

creases operational complexity and cost but also makes it difficult to enforce uniform security policies, prioritize remediation efforts, or even understand where sensitive data is being exposed.

AI-Driven Risk Expansion

The adoption of AI applications and agentic systems is fundamentally changing the threat model. While early discussions focused on stolen credentials or prompt injection, the more significant risk increasingly comes from a different direction. Agentic systems operate with delegated authority, act asynchronously, and chain API calls across multiple systems. Attackers can exploit this autonomy by manipulating agents into performing unsafe or unintended actions such as initiating payments, modifying privileges, or exporting sensitive data, all while staying within nominally authorized boundaries. Mitigating these risks requires guardrails that constrain autonomous behavior, including step-up authentication, contextual authorization checks, and optional human-in-the-loop approvals for high-risk actions.

In addition, agents are poorly served by authorization models based on static tokens and long-lived sessions. Instead, authorization must become continuous and event-driven, allowing access decisions to change mid-execution in response to evolving risk, policy updates, or revoked user consent. At the same time, there are clear limits to what API security can realistically enforce at the semantic level. Prompt injection remains largely

Chapter 1

a problem of model behavior and application architecture that depends on internal context and evolves faster than API security rules can adapt. API-level controls can reduce exposure through access control, rate limiting, and logging, but effective mitigation of agent manipulation must be implemented within the AI application itself.

Business Logic and Runtime Abuse

Attackers are increasingly targeting the intended functionality of APIs rather than exploiting classic technical vulnerabilities. Many of the most impactful threats described in the OWASP API Top 10, such as Broken Object Level Authorization (BOLA), Broken Function Level Authorization (BFLA), Excessive Data Exposure, and Unrestricted Resource Consumption, stem from weaknesses in business logic and access enforcement rather than flawed code. APIs are designed to expose structured access to data and functionality, and attackers exploit this by manipulating object identifiers, abusing legitimate functions, chaining requests in unexpected ways, or extracting more data than intended through overly permissive responses.

Traditional security controls focused on signature-based detection or perimeter defenses are largely ineffective against logic abuse, parameter tampering, mass assignment, or sequence manipulation. Runtime abuse also includes automated exploitation patterns such as credential stuffing, scraping, and token abuse, as well as denial-of-service attacks. Effective defense requires deep runtime visibility, behavioral baselining, and context-aware enforcement that understands how APIs are supposed to be used, not just how they are structured. Continuous monitoring and adaptive policies are essential to detect and mitigate these threats before they result in data exposure, fraud, or service disruption.

Scaling API Governance Across Fragmented Environments

Few organizations operate APIs within a single, unified environment. Instead, APIs are deployed across a heterogeneous mix of traditional API gateways, service meshes, Kubernetes ingress controllers, cloud-native platforms, on-premises legacy systems, and edge environments. Different teams often select different tools based on their own requirements, resulting in multiple gateways and enforcement points operating side by side. While this decentralization is often necessary to support scale and development flexibility, it significantly complicates visibility, governance, and security enforcement at the enterprise level.

In practice, the challenge is often less about defining the right policies and more about operationalizing them at scale. With hundreds or thousands of APIs, manual configuration and environment-specific management quickly become unmanageable. Security controls must be expressed as policy as code, integrated directly into development pipelines, and applied consistently across all environments. Automation becomes essential, including intelligent capabilities such as policy learning, automatic rule suggestions, and declarative enforcement layers that can adapt to dynamic infrastructure. Without a highly automated operating model, this fragmentation turns into an unsustainable operational burden.

Rising Compliance Pressure

The regulatory landscape is evolving rapidly, with APIs increasingly falling directly within the scope of data protection, financial, and AI governance frameworks. Regulations such as GDPR, HIPAA, PCI DSS, ISO 42001, and the EU AI Act require organizations to demonstrate control over how APIs expose, process, and transmit sensitive or regulated data. This includes enforcing strong authentication and authorization, validating API schemas, limiting data exposure, maintaining detailed logs, and preserving audit trails. As APIs become the primary integration layer for digital services and AI-driven workflows, regulators are treating them as critical control points for risk, accountability, and compliance.

The challenge for most organizations is enforcing and proving compliance requirements consistently across large, distributed API deployments. Effective compliance depends on automation, centralized visibility, and policy-driven enforcement that can be audited in near real time. API security and management platforms increasingly provide predefined policy templates, continuous compliance monitoring, and automated evidence generation to support audits. This shifts compliance from a periodic, manual exercise to an ongoing operational capability, reducing both risk and the cost of compliance over time.

The Solution

API security platforms increasingly serve as a unifying security and governance layer for digital services, rather than isolated point solutions. Their purpose is to protect APIs not only as technical interfaces, but as enforcement points where identity, policy, and application behavior converge. In complex enterprise environments, APIs often sit in front of legacy systems, cloud-native services, and external integrations at the same time, requiring consistent security controls.

A defining principle here is the tight coupling of access control and application-level protection. Instead of treating authentication, authorization, and threat prevention as separate disciplines, these platforms combine them into a shared runtime context. Identity signals, session state, and behavioral indicators can be evaluated, dynamically adapting access decisions and protection measures.

Modern platforms are designed to function across heterogeneous deployment models. Centralized governance is combined with decentralized enforcement, allowing security controls to be applied close to the workload while still being managed consistently. Support for traditional on-premises deployments, virtualized environments, containerized workloads, and managed services reflects the reality that API estates evolve incrementally rather than through clean-slate redesigns.

Taken together, the scope of such platforms extends beyond pure API protection. They aim to establish a coherent security fabric for digital services, where identity context can be propagated, policies enforced at run-

me, and risk signals shared across layers. This makes it possible to secure APIs at scale while accommodating regulatory requirements, architectural diversity, and the growing use of automation and non-human consumers.

Scope of Modern API Security

Discovery and Classification

Full visibility into your API landscape: own and third party, external or internal, sensitive or not

Design and Development

Security by design, coding best practices, testing, automation...

Access Control

Enforce consistent fine-grained access control to sensitive data

Data Validation

All data flows must conform to API contracts, policies, etc.

Analytics and Automation

Activity monitoring, anomaly detection, automated threat mitigation

Threat Protection

Resilience against malware, exploits, abuses, stolen credentials...

Transport Security

Traffic encryption, network isolation, DDOS prevention, anti-bot detection...

Chapter 2

How the solution works

API Discovery and Inventory

A continuously updated API inventory is the foundation of effective governance. Modern platforms use passive traffic analysis, integration with gateways and service meshes, Kubernetes inspection, and code repository scanning to discover APIs across all environments. This includes shadow APIs, zombie APIs, third-party integrations, and APIs used by automation or AI agents. Discovered APIs are enriched with metadata such as ownership, protocol, authentication method, and data sensitivity, providing the visibility needed to understand exposure and apply appropriate controls.

Security Posture Management

Posture management assesses APIs against security best practices, organizational policies, and frameworks, including but not limited to OWASP API Top 10. This covers evaluating authentication and authorization strength, encryption, schema consistency, and data exposure. APIs are typically scored based on risk and business impact, allowing teams to prioritize remediation efforts. Integration with CI/CD pipelines and issue-tracking systems ensures that posture findings translate into actionable improvements rather than static reports.

Threat Prevention and Runtime Protection

Runtime protection enforces policies on live API traffic, validating requests against expected behavior, applying rate limits, and detecting anomalous usage patterns. This helps defend against common API threats such as BOLA, excessive resource consumption, credential abuse, and automated business logic attacks. Enforcement mechanisms increasingly support adaptive responses, which allows policies to be updated or applied dynamically as conditions change.

Identity-centric Access Control

Modern APIs require access control models that go beyond network-centric assumptions. Identity-centric access control bases decisions on the authenticated identity and its context, covering both human users and non-human identities such as services, bots, and AI agents. Different identity types can be subject to different trust levels and authorization policies. In AI-driven scenarios, this enables fine-grained, continuous authorization for autonomous agents, helping verify their legitimacy, constrain delegated authority, and support step-up authentication or human-in-the-loop approval for sensitive actions.

API Analytics and Security Intelligence

API analytics provide visibility into traffic patterns, performance metrics, error rates, and usage trends across all environments. Security intelligence builds on this data by applying behavioral analysis to detect abnormal usage, suspicious call sequences, or emerging abuse patterns. Integrated Machine Learning (ML) can help identify patterns indicative of business logic abuse, credential stuffing, or AI-specific attacks. Insights from analytics feed back into policy tuning, posture improvement, and compliance reporting, creating a continuous feedback loop.

Compliance and Audit Readiness

Compliance capabilities are embedded directly into modern API platforms. Features such as data classification, role-based access control, immutable logging, and audit-ready reporting support regulatory and standards requirements, which is especially relevant for European countries. Continuous monitoring and automated evidence generation reduce reliance on manual audits and help organizations demonstrate ongoing compliance, even in highly regulated industries.

Developer and Ecosystem Enablement

Developer enablement remains a critical component of API platforms. Developer portals, API catalogs, documentation, and self-service onboarding help teams and partners consume APIs efficiently while staying within governance boundaries. Integration with CI/CD pipelines ensures that security and compliance controls are applied early. For external or monetized APIs, enablement extends to controlled onboarding, usage tracking, and abuse prevention, balancing ecosystem growth with protection.

Top Use Cases

AI-Driven API Security (Protecting LLM and Agentic Workflows)

APIs are now the control plane for AI workloads, integrating LLMs and agentic systems into business processes. They introduce new security risks, such as prompt injection, which can cause models to produce harmful or unintended output; data exfiltration through chained prompts or embedded requests; and abuse of model resources through excessive token consumption. API security platforms help mitigate exposure and blast radius of these risks. They implement schema validation to filter malicious or malformed requests, semantic input filtering to sanitize prompts, token governance to prevent resource exhaustion, and access controls to restrict API exposure to authorized users and agents. For agentic systems, solutions provide monitoring of chained API calls to ensure that automated workflows do not violate policy or access sensitive systems without authorization. This is important for organizations that are embedding AI into customer-facing services, where misuse could cause operational, financial, or reputational harm.

API Security Posture Management (Complete API Discovery & Risk Assessment)

In modern complex IT environments, undocumented APIs are inevitable. Without visibility, these shadow APIs remain unmanaged, creating hidden attack surfaces. Zombie APIs that persist after deprecation pose equally serious risks, often retaining access to sensitive systems. API Security Posture Management provides continuous discovery across all environments, automatically classifying APIs by sensitivity, exposure, and compliance requirements. Posture assessments identify weak authentication, missing encryption, excessive permissions, and schema drift. Risk scores are calculated based on both technical severity and business impact, which enables the prioritization of remediation efforts. Integration with ticketing systems ensures that high-risk findings are tracked to resolution. For organizations with hundreds or thousands of APIs, posture management is the only way to maintain a defensible security position.

Full Lifecycle API Security (Threat Prevention, Runtime Detection, and Response)

API security must be embedded across the entire lifecycle, from initial design through production runtime to deprecation. At the design stage, API contracts are validated, security quality gates are applied in continuous delivery pipelines, and static analysis tools scan for vulnerabilities. In production, runtime detection continuously monitors API traffic for anomalies, applying behavioral baselines to detect attacks that bypass static defenses. Threats such as BOLA, credential abuse, and automated business logic attacks are detected in real time. Response capabilities integrate with security analytics platforms to trigger alerts, apply policy updates, or block malicious traffic automatically. Post-incident forensics provide insights to prevent recurrence. This full lifecycle approach ensures that APIs remain secure throughout their entire life, adapting to new threats as they emerge.

Chapter 3

Unified API Fabric (Management of Hybrid, Multi-Cloud, and Edge Environments)

Enterprises now operate APIs across a wide variety of platforms: gateways, service meshes, Kubernetes clusters, cloud providers, on-premises systems, and edge deployments. Managing policies, security controls, and governance across these environments is a major challenge. A unified API fabric provides a central control plane for policy definition and monitoring, with enforcement distributed across all environments. Policies for authentication, rate limiting, compliance, and threat protection are defined once and applied consistently everywhere. For edge deployments, local enforcement nodes apply policies close to the workload, reducing latency while maintaining consistency. This approach ensures that security and compliance are uniform, even in highly decentralized environments.

API Monetization & Ecosystem Enablement

As APIs become products, organizations must balance business with security. Monetized APIs often expose business-critical capabilities to external developers, partners, or marketplaces. This creates new risks of abuse or exploitation by malicious actors. Secure monetization requires strong access control, usage metering, SLA enforcement, and fraud prevention. Developer portals provide controlled onboarding, documentation, and sandbox environments to enable ecosystem growth without compromising security. Monitoring and analytics ensure that monetized APIs perform as promised while protecting revenue streams.

Compliance-Driven API Governance

APIs handling sensitive or regulated data are now explicitly in the scope of compliance frameworks. API governance must ensure that access controls, schema validation, logging, and audit trails are enforced consistently. Platforms provide compliance dashboards aligned with frameworks like GDPR, HIPAA, PCI DSS, ISO 42001, and the EU AI Act. Automated evidence generation simplifies audit preparation. By embedding compliance into the API governance model, organizations can demonstrate adherence to regulations without building separate, costly compliance infrastructure.

Top Functional Selection Criteria

API Lifecycle Management

Full coverage from design and testing through deployment, operations, and retirement. Includes contract-first API design, policy templates, automated conformance testing, CI/CD integration, and lifecycle governance. Strong solutions enable smooth transitions between lifecycle stages without introducing security gaps.

Deployment and Integration

Solutions are deployed across on-premises, cloud, hybrid, and edge environments. Integration with multiple API gateways, service meshes, Kubernetes environments, and CI/CD pipelines is critical for maintaining consistent policy enforcement, unified observability, and centralized governance in complex, hybrid environments.

Developer Portal and Tools

Developer-facing capabilities include documentation, subscription management, streamlined onboarding, and self-service key provisioning to reduce friction for development teams. Integration with DevOps pipelines ensures that security, compliance, and governance controls are built directly into the development lifecycle. Additional tooling for testing, sandboxing, and versioning can further improve API quality, usability, and security posture.

Identity and Access Control

While static API keys still have their use in certain scenarios, considering them a viable method of authentication or access control is no longer a sensible idea. Support for diverse identity types and standards such as OAuth2, OpenID Connect, SAML, and Verifiable Credentials ensures broad compatibility and interoperability. Context-aware, runtime authorization delivers fine-grained and consistent access governance across all environments. Capabilities such as just-in-time provisioning, policy-based entitlements, and adaptive risk-based controls strengthen overall identity assurance and access security.

API Vulnerability Management

Automated discovery and validation of APIs against specifications, industry security best practices, and the OWASP API Top 10 helps ensure robust protection from common weaknesses. Advanced vulnerability scanning, misconfiguration detection, and continuous monitoring identify potential exposures before they are exploited. Business-risk-based prioritization ensures that critical issues are addressed promptly, with actionable remediation guidance to reduce security debt.

Analytics and Security Intelligence

Continuous monitoring of API traffic enables anomaly detection and behavioral analytics to identify anomalies that may indicate misuse or attack attempts. Integration with SIEM and XDR platforms enhances incident correlation and response by combining API events with broader enterprise security intelligence. Visualization dashboards provide actionable insights into performance trends, usage metrics, and the overall API security posture.

Chapter 4

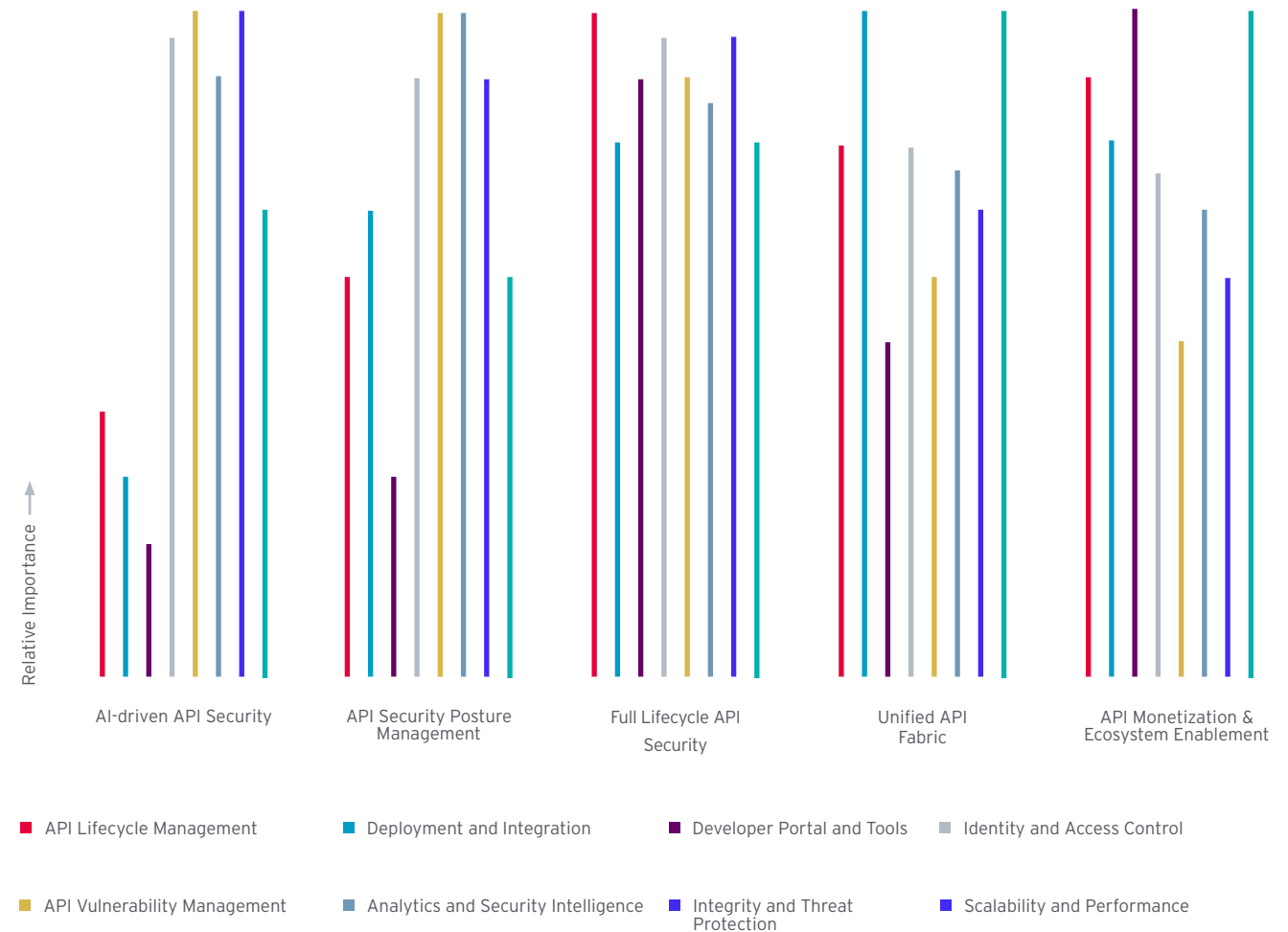
Integrity and Threat Protection

Protection mechanisms defend against transport-level and protocol-specific threats, including injection attacks, schema violations, business logic abuse, and bot-driven activity. Proactive blocking combined with adaptive run-time defenses ensures that APIs can respond effectively to evolving and sophisticated attack techniques.

Scalability and Performance

The ability to scale horizontally and vertically to meet growing demand, maintain low latency, and remain resilient during peak loads or denial-of-service conditions. Advanced load balancing, caching, and performance optimization techniques ensure predictable behavior under varying workloads.

Use Case / Capabilities Mapping



Vendor Spotlight

Founded in 1984 and headquartered in Zürich, Switzerland, Ergon Informatik AG is a long-established software engineering company with a strong and consistent focus on cybersecurity. Its flagship product suite, the Airlock Secure Access Hub, combines identity and access management with comprehensive web application and API protection into a unified platform. Serving customers in more than 30 countries, Ergon has a particularly strong presence in the DACH region while steadily expanding into markets such as the Middle East and APAC.

Ergon's strategy is centered on the convergence of identity management and application and API protection into a cohesive security fabric. Instead of treating access control, authentication, and threat prevention as isolated disciplines, Airlock unifies these capabilities through continuous trust evaluation, real-time signal exchange between access and application layers, and decentralized enforcement models.

The Airlock Secure Access Hub consists of several tightly integrated components. Airlock Gateway provides WAF and API security functionality and is typically deployed as a virtual appliance, while Airlock Microgateway addresses cloud-native environments through sidecar and Kubernetes Gateway API modes. Airlock IAM serves as the identity backbone of the platform and is available both on-premises and as a SaaS offering.

Airlock's approach to API security differs from platforms that emphasize broad API discovery and posture management. It assumes that APIs entering production are

explicitly defined, documented, and intentionally exposed. Its design philosophy prioritizes enforcing security and access control at runtime against declared API contracts, with a strong emphasis on identity context, authorization, and behavioral enforcement. The platform supports a broad range of API standards, including REST, SOAP, GraphQL, gRPC, XML-RPC, and others.

Protection starts with strong API specification enforcement, including request and response validation, JSON schema whitelisting, and protocol-level filtering. Legacy backends can be protected through reverse proxying and identity augmentation without application changes. In continuous delivery environments, Airlock Microgateway enables close integration with DevSecOps workflows. API discovery and inventory are not part of Airlock's current scope. The platform is intentionally focused on runtime security, rather than proactive vulnerability testing.

For monitoring and analytics, Airlock provides extensive logging and visualization through integrations with Kibana and Grafana, along with prebuilt dashboards for usage patterns, policy violations, and behavioral anomalies. The Anomaly Shield module applies machine learning techniques such as session-based profiling, automatic model retraining, and transfer learning to provide immediate protection for newly deployed services.

The platform delivers strong integrity and threat protection capabilities and covers the OWASP API Top 10 risks through runtime enforcement and access control.

It includes built-in defenses against denial-of-service attacks, bot abuse, XML, JSON, and SOAP-based attacks, and various forms of protocol misuse. Enforcement can be applied centrally or at the edge, depending on the deployment model. Additional protection layers are provided through integration with BrightCloud threat intelligence and optional ICAP-based malware scanning.

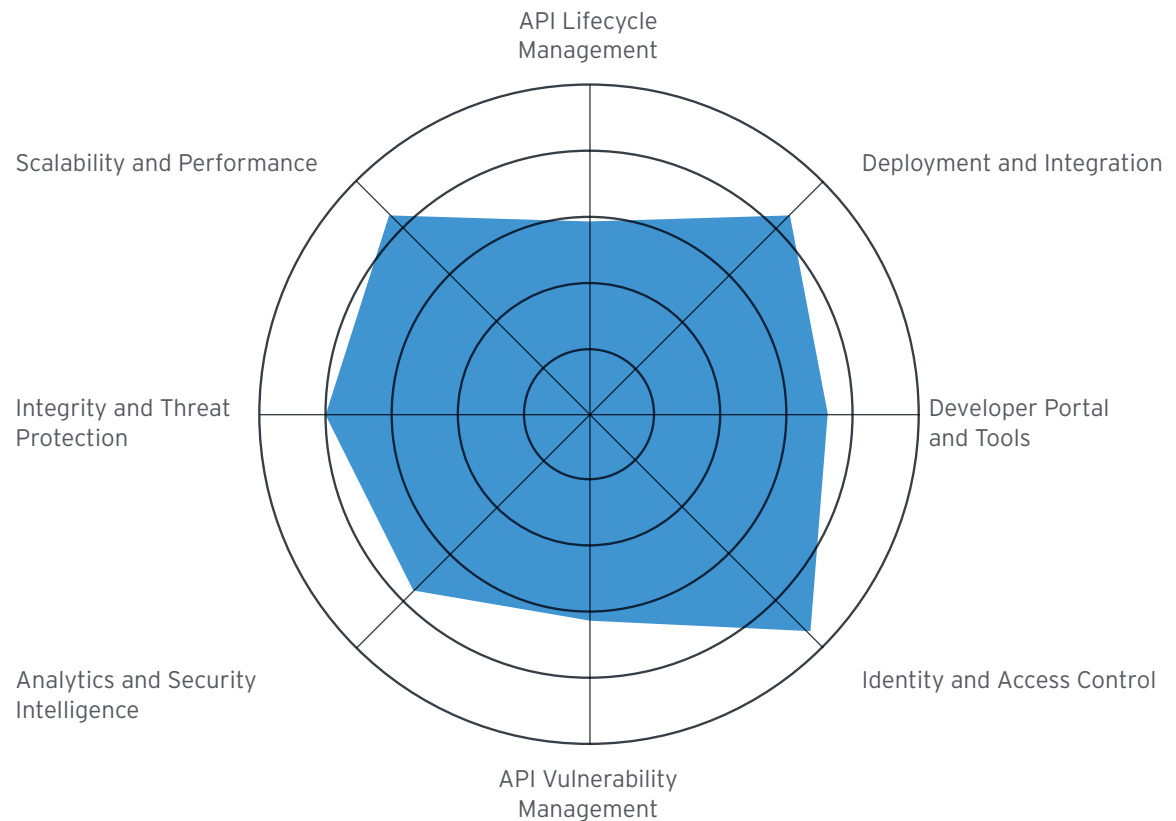
Deployment flexibility is one of Airlock's key strengths. It supports traditional on-premises deployments, public and private cloud environments, containerized workloads, and increasingly SaaS-based consumption models. Hybrid trust architectures are supported through advanced token exchange mechanisms that allow identity context to be propagated across security zones and presented in backend-native formats, enabling Zero Trust enforcement without disrupting legacy systems.

Airlock's primary differentiator lies in the deep integration between IAM and API security capabilities. Unified signal exchange, advanced identity propagation, and runtime authorization enforcement allow organizations to adapt access and protection decisions dynamically based on user identity, behavior, and risk. Comprehensive support for identity standards and post-quantum cryptography combined with strong ML-based threat detection capabilities further strengthen the platform's position. Challenges remain around the relatively small but growing global partner ecosystem, the lack of proactive vulnerability analysis, and the absence of generative AI-based security functions.

Chapter 5

Overall, Ergon offers a comprehensive and mature API security solution through its Airlock suite, combining identity-aware access controls with advanced runtime security capabilities in a unified Zero Trust architecture. The solution is particularly well-suited for mid- to large-scale enterprises with complex hybrid infrastructures and strict regulatory or data sovereignty requirements, and it is recognized as a product leader for its usability, deployment flexibility, and deep integration of identity and application security.

Overview of Vendor Capabilities



Considerations

Architectural Considerations

Modern API security and management platforms are designed to operate across diverse environments, from centralized gateways to distributed, edge-native deployments. Many solutions can be delivered as SaaS or at least as a managed service, but flexible options often include self-managed or hybrid deployment models to accommodate data sovereignty and regulatory constraints.

Deployment architecture must account for scalability, latency, and availability requirements. Some organizations may opt for a cloud-hosted control plane with distributed enforcement nodes running in multiple regions, while others will need fully on-premises or private cloud deployments to satisfy compliance mandates. Increasingly, edge-native enforcement is becoming a consideration for APIs that power latency-sensitive applications in IoT, retail, or industrial automation.

Integration is another key aspect of API platforms. Solutions should integrate natively with API gateways, service meshes, and orchestration frameworks already in use. Integration with CI/CD pipelines ensures that security checks are embedded into the API development process. Interoperability with IAM/IAG, SIEM, SOAR, and XDR platforms extends security visibility and incident response capabilities across the broader enterprise ecosystem. A strong integration model allows API security to function as part of a unified security and governance fabric, rather than as an isolated control point.

Pre-deployment considerations

Before deploying an API Security and Management solution, organizations should address a set of fundamental readiness questions.

What are your primary use cases?

This is the key question that determines the entirety of your future API strategy. Even if you aim to cover all your bases in the future, carefully choosing your priorities early will massively influence your long-term timeframes, budgets, and choice of vendors.

What is the current state of the API inventory?

Understanding the scale and complexity of your API environment, including shadow and zombie APIs, is essential for planning integration and onboarding into the platform.

Which deployment environments are in scope?

Consider whether the solution needs to support APIs running on multiple clouds, in service meshes, in Kubernetes clusters, on legacy on-premises systems, or at the edge. This will define the architecture and deployment model required.

What security policies and standards are already in place?

Review existing authentication, encryption, logging, and compliance policies to ensure the platform aligns with current practices, such as industry-specific technical standards and corporate security policies.

How will the platform integrate with existing business processes?

Consider how API security and governance will fit into development lifecycles, incident response processes, and compliance workflows. Integration with CI/CD, ITSM, and governance processes will help ensure adoption and reduce operational friction.

Who will own the platform and its processes?

Determine whether API security will be owned by a dedicated API team, the security operations center, the DevOps function, or a cross-functional governance team. Clear ownership will accelerate adoption and ensure accountability.

Chapter 6

Questions to Ask Vendors during RFPs

What are your key differentiators?

- Understanding the key differentiators between the vendors is essential. Ask the vendor what their unique capabilities are, what are the primary areas their development is focusing on, what specific industry verticals they are serving, etc.
- Airlock's response:
Airlock's primary differentiator is the tight integration of API security and identity and access management within the Airlock Secure Access Hub. It combines both capabilities into a single, cohesive architecture with shared policy enforcement and runtime context. While each component can be deployed independently, their integration enables consistent access control and security enforcement across heterogeneous API environments, reducing operational complexity and policy gaps.

How does your solution protect LLMs or agentic AI systems?

- This question assesses whether the vendor has capabilities to address emerging AI-specific threats, such as prompt injection, context manipulation, and token abuse, which are increasingly relevant for modern applications.
- Airlock's response:
Airlock protects AI systems primarily at the API and access layer, rather than by inspecting or filtering AI prompts or model interactions. It does not position itself as a prompt inspection or model governance solution but focuses on controlling how users and AI agents authenticate, authorize, and consume APIs. Core protections include abuse prevention mechanisms such as rate limiting and throttling, as well as

strong authentication and authorization for both human users and AI agents. These controls limit token abuse, excessive API usage, and unauthorized access regardless of whether the caller is a traditional application or an AI-driven agent.

For agentic AI scenarios, Airlock enforces established delegation and identity patterns. It supports agentic On-Behalf-Of flows using dual-identity tokens based on RFC 8693 Token Exchange, enabling traceability of both human and non-human identities. It also applies a Proof of Continuity model, evaluating whether a presented token is sufficient for a requested operation and triggering token exchange when stronger authority is required.

In addition, Airlock can support trust-spanning identity frameworks through decentralized identifiers and verifiable credentials, enabling identity, delegation, and trust relationships across organizational boundaries while keeping enforcement anchored at the API and access control layer.

How do you discover and manage shadow and zombie APIs across environments?

- Full visibility is essential for security. This question evaluates the platform's ability to discover unmanaged APIs across fragmented environments, ensuring no API remains outside governance.
- Airlock's response:
Airlock does not position itself as a broad API discovery or inventory tool. Instead, it prevents the exposure and use of shadow and zombie APIs within the scope of traffic that is filtered through Airlock. By operating in a secure-by-default mode, APIs that are not explicitly published or allowed are not reachable via the Airlock gateway.
Enforcement goes beyond basic attack detection and filtering. Airlock can enforce OpenAPI specifications at runtime, ensuring that only documented endpoints and methods are accessible. API calls targeting unde-

ocumented, unpublished, or deprecated endpoints are rejected, effectively eliminating the risk of shadow or zombie APIs on managed traffic.

What runtime threat detection capabilities are available?

- Business logic and runtime attacks often bypass static defenses. The vendor's response should demonstrate behavioral analytics, sequence enforcement, and integration with both SOC operators and developer teams for rapid mitigation.
- Airlock's response:
Airlock provides runtime threat detection through its Anomaly Shield capability, which applies machine learning analysis to API and application sessions. Anomaly Shield evaluates session behavior over time to identify anomalous patterns, including unexpected call sequences and interaction flows. This approach enables Airlock to detect suspicious activity early in the attack lifecycle, including reconnaissance phases where attackers use automated tools to probe APIs and applications for weaknesses.
Detected anomalies can be forwarded to SIEM platforms, allowing SOC teams to monitor, investigate, and respond using established security operations processes. Additionally, Airlock can take protective actions at runtime, such as blocking or deterring malicious sessions, supporting rapid mitigation of business logic and behavioral attacks as they occur.

Chapter 6

How do you support distributed, hybrid, and edge deployments with consistent governance?

- Distributed API environments require consistent policy enforcement across diverse deployment types. The answer to this question reveals whether the platform can act as a unifying API fabric with centralized governance and decentralized enforcement.
- Airlock's response:
Airlock supports complex deployments by combining centralized identity and policy decisions with decentralized enforcement across different runtime environments. At the edge, Airlock Gateway provides protection for internet-facing and network-edge deployments. In cloud-native and containerized environments, Airlock Microgateway enables enforcement directly in front of individual services or applications. Airlock IAM provides unified authentication and authorization, enforcing strong identity controls across deployment models. The components are integrated using OpenID Connect to authenticate and authorize users and clients, with entitlements enforced before requests reach backend APIs. This allows legacy backends to be protected with modern standard-based access control without changes to the backend systems themselves. In addition, Airlock supports Token Exchange to establish separate security zones and to ensure that only authorized clients can access services across trust boundaries.

What capabilities exist for API productization and monetization?

- For monetized APIs, security must be embedded into productization. This question clarifies the vendor's capabilities for secure ecosystem growth, usage metering, and fraud prevention.
- Airlock's response:
Airlock supports basic API productization and monetization use cases by enforcing client-specific usage

controls at runtime. Customers can define throttle limits and usage plans on a per-API client basis, helping to prevent abuse and support differentiated service levels.

In addition, Airlock exposes detailed usage and traffic data through metrics endpoints. This telemetry can be consumed by external systems for billing, reporting, or fraud analysis, while Airlock itself remains focused on enforcing usage limits and access controls.

How does your platform assist with regulatory compliance?

- Compliance is mandatory in regulated industries. The answer should cover built-in reporting, policy templates, automated evidence generation, and continuous compliance monitoring.
- Airlock's response:
Airlock supports regulatory compliance through its long-standing use in highly regulated industries, particularly in the financial sector, where it has been deployed across multiple jurisdictions for more than two decades. This experience has shaped the platform's ability to meet regulatory expectations around access control, security enforcement, and auditability. Rather than offering prescriptive compliance templates, Airlock provides security and access control capabilities that support regulatory obligations in the markets where the solution is deployed, with compliance implementation driven by customer needs.

What is on your short-term and long-term roadmap?

- The list of features on a vendor's roadmap is a good indicator of their innovativeness and the vendor's ability to support emerging use cases. What features are they planning to deliver in the next 3 months or next year?
- Airlock's response:
Our roadmap currently focuses on self-sovereign

identity technologies, which form the foundation for upcoming national digital identity initiatives such as the EU Personal Identity Wallet and the Swiss E-ID. Airlock already supports the relevant public beta infrastructures for these initiatives.

As the national infrastructures become operational, Airlock will be able to integrate with and support both the EU and Swiss digital identity ecosystems. These developments reflect our focus on enabling secure, standards-based identity and access scenarios for emerging cross-border use cases.

Can I speak to some reference customers?

- Speaking to an existing customer, especially from a similar industry and region as your organization, can reveal a lot of information about the vendor's background and practices. This includes not just their relationship with the vendor but valuable insights into the product's strengths and shortcomings.
- Airlock's response:
Yes, of course. Check out our web site with our success stories and contact us so we can help you in setting up a meeting.

Chapter 6

Recommendations

Organizations should approach API security and management strategically, with an understanding that APIs now represent both a primary driver for innovation and one of the most significant enterprise risk surfaces. A successful implementation begins with recognizing that security and management cannot be separated: platforms that unify both capabilities across the full API lifecycle will deliver the most consistent protection and governance.

Visibility should be the starting point for any evaluation. Without a complete, continuously updated inventory of APIs, no security or compliance program can operate effectively. Once visibility is established, the chosen platform must provide security controls embedded throughout the API lifecycle, from design-time validation to runtime enforcement and post-incident forensics.

It is also important to view AI security as an immediate requirement, not a future consideration. However, the primary value of API security in AI architectures lies in governing access, authorization boundaries, and permitted actions, rather than attempting to control model behavior or semantic interpretation. APIs are the integration layer for LLMs and agentic systems, and securing these endpoints is essential for preventing data leakage, model misuse, and prompt injection. Platforms that can enforce contextual access controls, constrain delegated authority, and monitor chained API interactions will be far better positioned to adapt to the rapidly evolving AI landscape.

Organizations should further prioritize identity-centric access control as a foundational design principle for API security. APIs are increasingly consumed by non-human identities such as AI agents operating with delegated authority. Strong authentication, federation, and fine-grained authorization must be enforced consistently across web applications and APIs, independent of where they are deployed. By treating identity as the primary control plane and enforcing policies at gateways and application entry points, organizations can unify access governance and reduce the friction between IAM and API security stacks.

Enterprises should move away from static, session-based access decisions toward continuous authentication and authorization. Access decisions must be re-evaluated dynamically based on contextual signals, policy updates, or revoked consent, particularly in long-running or asynchronous workflows. For high-impact API activities, especially those initiated by AI workflows or agents, organizations should require explicit safeguards such as step-up authentication or human-in-the-loop approval to prevent authorized but unsafe execution. These capabilities are also becoming essential for limiting blast radius of API-focused attacks.

Given the diversity of modern API environments, enterprises should prioritize platforms capable of managing multiple gateways, service meshes, and deployment models across hybrid, multi-cloud, and edge infrastructures. This requires architectures that separate centralized policy definition and decision-making from enforcement, allowing controls to be applied locally at gateways, microgateways, or edge nodes while maintaining consistent governance. These architectures ensure uniform governance without sacrificing performance or resilience to forcing all traffic through a single chokepoint.

Finally, compliance must be treated as a core evaluation criterion. The ability to enforce governance policies, generate audit reports, and demonstrate ongoing adherence will reduce both the cost and complexity of meeting regulatory obligations. By selecting a platform that aligns security, operational efficiency, and compliance, organizations can manage risk more effectively while still focusing on the business value of their investment.

Related Research

Leadership Compass

- API Security and Management
- Web Application and API Protection
- Access Management
- Passwordless Authentication B2C

Advisory Note

The Role of APIs for Business

Whitepaper

- The Dark Side of the API Economy
- How to make CIAM a Success

Executive View

Airlock Secure Access Hub for Applications and APIs

Copyright

©2025 KuppingerCole Analysts AG. All rights reserved. Reproducing or distributing this publication in any form is prohibited without prior written permission. The conclusions, recommendations, and predictions in this document reflect KuppingerCole's initial views. As we gather more information and conduct deeper analysis, the positions presented here may undergo refinements or significant changes. KuppingerCole disclaims all warranties regarding the completeness, accuracy, and adequacy of this information. Although KuppingerCole research documents may discuss legal issues related to information security and technology, we do not provide legal services or advice, and our publications should not be used as such. KuppingerCole assumes no liability for errors or inadequacies in the information contained in this document. Any expressed opinion may change without notice. All product and company names are trademarks™ or registered® trademarks of their respective holders. Their use does not imply any affiliation with or endorsement by them.

KuppingerCole Analysts supports IT professionals with exceptional expertise to define IT strategies and make relevant decisions. As a leading analyst firm, KuppingerCole offers firsthand, vendor-neutral information. Our services enable you to make decisions crucial to your business with confidence and security.

Founded in 2004, KuppingerCole is a global, independent analyst organization headquartered in Europe. We specialize in providing vendor-neutral advice, expertise, thought leadership, and practical relevance in Cybersecurity, Digital Identity & IAM (Identity and Access Management), Cloud Risk and Security, and Artificial Intelligence, as well as technologies enabling Digital Transformation. We assist companies, corporate users, integrators, and software manufacturers to address both tactical and strategic challenges by making better decisions for their business success. Balancing immediate implementation with long-term viability is central to our philosophy.

For further information, please contact clients@kuppingercole.com.

KuppingerCole Analysts AG

Wilhelmstraße 20-22
65185 Wiesbaden | GERMANY

P: +49 | 211 - 23 70 77 - 0

F: +49 | 211 - 23 70 77 - 11

E: clients@kuppingercole.com.

[kuppingercole.com](https://www.kuppingercole.com)