

WHITEPAPER

API Catalog

Enterprise Discovery,
Governance, Security, ROI
and AI Readiness

Enterprise Architecture paper
for CXO, Digital, Security, Platform
and Integration Leaders



Index Page

1. Executive Summary for CXO	3
2. Enterprise Problem Statement	4
3. API Catalog Definition and Scope	5
4. RFC 9727 and Standardized API Discovery	6
5. Business Capability View of API Catalog	7
6. Reference Architecture	8
7. Metadata and Ownership Model	9
8. API Product Thinking and Discoverability	10
9. Governance and Policy Enforcement	11
10. Security and Compliance Controls	12
11. Lifecycle Management	13
12. Comparison Table: Catalog, Registry, Gateway, Portal	14
13. Benefits and Measures	15
14. ROI Model and Financial Measurement	16
15. CXO Metrics Dashboard	18
16. AI Readiness and Machine-Consumable APIs	19
17. Implementation Roadmap	20
18. Operating Model and Roles	21
19. CXO Decision Framework and Conclusion	22
20. Annexure - References and Links	23

1. Executive Summary for CXO

Most enterprises have more APIs than they can confidently govern. APIs sit in gateways, repositories, codebases, SaaS platforms, cloud services and partner programs. The business impact is not only technical complexity; it is executive uncertainty. Leadership needs to know what APIs exist, what business capability they expose, who owns them, how secure they are and whether they are being reused.

An API Catalog provides the answer. It is the governed portfolio system for API assets. It connects API identity, business purpose, technical contract, ownership, lifecycle, security, compliance, usage and adoption into one searchable source of truth. The uploaded master paper already positions API Catalog as a centralized, living repository combining discovery, reuse, governance, lifecycle, security and AI readiness.

RFC 9727 adds strategic relevance by defining the api-catalog well-known URI and link relation for automated discovery of published APIs. In practical terms, this means API Catalog is no longer only an internal convenience. It has become part of a global movement toward standardized API discovery.

CXO Question	API Catalog Answer	Measurable Signal
What APIs do we have?	Central inventory with search and business metadata	Catalog coverage rate
Can teams reuse APIs?	Business capability tags, documentation and consumer visibility	API reuse rate
Are APIs safe?	Security metadata, owner, auth method, lifecycle and quality score	Security score and shadow API count
Are APIs valuable?	API product view, usage, SLA, consumer and adoption tracking	Adoption and value metrics
Are APIs AI-ready?	Machine-readable contracts and intent metadata	AI readiness score

CXO Message

A CXO should view API Catalog as digital asset governance, not as another documentation website.

2. Enterprise Problem Statement

The API problem is rarely the absence of APIs. The real problem is the absence of trusted API knowledge. When teams cannot find existing APIs, they rebuild. When security cannot see APIs, risk grows. When owners are unclear, incident response slows. When lifecycle is not tracked, old APIs remain active after the business no longer needs them.

The uploaded paper highlights several recurring enterprise issues: duplicate API development, shadow APIs, zombie APIs, slow onboarding, weak ownership, poor documentation and audit difficulty. These are not small development concerns. They directly influence cost, risk, speed and customer experience.

Problem	Observed Enterprise Pattern	Business Impact	Catalog Response
Duplicate APIs	Multiple teams solve the same integration need independently	Waste of design, build, test and support effort	Search-before-build and reuse visibility
Shadow APIs	APIs run without registration or review	Unknown attack surface and audit exposure	Discovery, mandatory registration and owner assignment
Zombie APIs	Old endpoints remain active after replacement	Operational cost and unpatched risk	Lifecycle status and retirement workflow
Slow onboarding	Developers depend on people and old documents	Delayed delivery and poor experience	Readable docs, examples and owner contacts
Manual audit	Evidence lives across teams and systems	High compliance effort	Central policy and metadata evidence

Enterprise Problem

API Catalog converts scattered API knowledge into governed enterprise memory.

3. API Catalog Definition and Scope

An API Catalog is a searchable, governed enterprise record of APIs and API-like digital capabilities. It is not just a list of runtime endpoints. It should describe what an API does, why it matters, who owns it, what contract defines it, where it runs, who consumes it, how secure it is and what lifecycle state it is in.

A mature catalog serves multiple audiences. Executives see API portfolio health. Architects see standards and domain alignment. Developers discover and reuse APIs. Security teams see exposure and control evidence. Platform teams see operational readiness. Product teams see adoption and monetization potential.

Scope Boundaries

API Catalog Includes	API Catalog Should Not Become
API identity, purpose, and business domain	A dumping ground for every runtime log
OpenAPI, AsyncAPI, GraphQL or other contract references	A replacement for source code repositories
Ownership, support model, and lifecycle status	An informal team wiki without governance
Security, data sensitivity, and compliance metadata	A place to store secrets or credentials
Consumer, dependency, and usage evidence	A one-time spreadsheet inventory

A Simple Definition

API Catalog is the enterprise product register for APIs - searchable like a marketplace, governed like a control system and measured like a business portfolio.

4. RFC 9727 and Standardized API Discovery

RFC 9727 is a standards milestone for API Catalog. It defines the api-catalog well-known URI and link relation. The stated intent is to facilitate automated discovery with the help of integrating / connecting to all systems and usage of published APIs. As of now there is no standard integrating / connecting way although, A request to the api-catalog resource returns a document providing information about and links to the publisher's APIs.

For business leaders, the significance is straightforward: API discovery is becoming standardized. This is relevant for partner ecosystems, external developer platforms, regulated data exchange, industry networks and future AI-agent consumption. A catalog that aligns to RFC 9727 is easier to reason about, easier to discover and easier to integrate into automation patterns.

RFC 9727 Element	Business Meaning	Architecture Implicationv
/.well-known /api-catalog	Predictable discovery location for catalog-aware consumers	Tools can search for catalog information consistently
api-catalog link relation	Standard relationship pointing to catalog document	Catalog discovery can be automated
API catalog document	Information and links about published APIs	Catalog data must be maintained and trustworthy
Metadata and links	Context about APIs, descriptions and related resources	API records should include meaningful business and technical links
Operational considerations	Maintenance, scalability and integration matters	Catalog must be treated as operational infrastructure

Standards Message

RFC 9727 does not replace governance.
It gives governed API discovery a standard surface.

5. Business Capability View of API Catalog

API Catalog should be understood as a business capability map, not only a technical library. Well-described APIs expose business functions such as customer lookup, order status, payment initiation, shipment tracking, case creation or identity validation. When APIs are tagged by business domain and capability, the catalog becomes meaningful to product, architecture and leadership teams.

This capability view is essential for reusability. Developers do not always search by endpoint name. They search by business intent. A mature catalog enables queries such as “find an API to retrieve customer profile” or “which APIs expose inventory availability.”

Capability Lens	Catalog Metadata Needed	CXO Value
Business domain	Domain, subdomain, capability, product owner	Portfolio visibility by business area
Consumer model	Internal, partner, public, AI-agent or application consumer	Exposure and access strategy
Value stream	Order, customer, supply chain, finance, HR, service	Business-process alignment
Data sensitivity	Public, internal, confidential, personal, regulated	Risk-based governance
Reusable product	Description, version, SLA, docs, usage and owner	API product management

Capability View

The catalog becomes powerful when business capabilities are searchable, not just technical endpoint names.

6. Reference Architecture

A production-grade API Catalog is a platform capability that connects experience, governance, and runtime layers. If the catalog is isolated from gateways, CI/CD, identity, and observability, it quickly becomes stale. If it is integrated, it becomes the trusted API system of record.

API Catalog Reference Architecture



Architecture Principles

- Catalog registration begins at design time and continues through retirement.
- API contracts such as OpenAPI or AsyncAPI are linked as authoritative evidence.
- Gateway and runtime telemetry should feed operational usage and health data back to the catalog.
- Identity and access systems should enforce role-based visibility and modification rights.
- Security and quality policies should be applied as promotion gates, not afterthoughts.

7. Metadata and Ownership Model

The quality of an API Catalog is determined by metadata quality. A catalog with missing owners, vague descriptions and outdated contracts will not be trusted. A catalog with consistent metadata can become the authoritative asset register for API governance and reuse.

The uploaded API Catalog paper provides a strong metadata structure covering identity, technical fields, ownership, lifecycle, SLA, security, compliance, and AI readiness. The following model simplifies that structure into CXO-friendly groups.

Metadata Group	Example Fields	Why it Matters
Identity and business context	API name, ID, domain, capability, business description	Makes APIs discoverable by business intent
Technical contract	Specification type, contract link, protocol style, version	Makes APIs understandable and automatable
Ownership	Business owner, technical owner, support team, reviewer	Creates accountability
Lifecycle and SLA	Status, release date, deprecation date, SLA tier	Controls risk and consumer expectations
Security and compliance	Data sensitivity, auth method, scopes, compliance flags, trace header	Supports risk and audit decisions
AI readiness	Machine-readable intent, docs, examples, tags, contract quality	Prepares APIs for agent consumption

Metadata Principle

No owner, no contract, no lifecycle status means no trust. Metadata is the foundation of catalog credibility.

8. API Product Thinking and Discoverability

An API becomes an enterprise product when it has a clear consumer, purpose, owner, lifecycle, service expectation, documentation and adoption measure. API Catalog is the place where that product identity is made visible.

For CXO audiences, product thinking matters because it shifts API investment from project-by-project integration into reusable digital capability creation. A project API may solve one demand. A cataloged API product can solve many future demands.

API Product Attribute	Catalog Representation	Measure
Clear purpose	Plain-English business description	Discovery success rate
Known owner	Business and technical owner	Owner completeness %
Reliable contract	OpenAPI / AsyncAPI / GraphQL specification link	Contract coverage %
Consumer readiness	Docs, usage guide, examples and access model	Documentation completeness %
Service commitment	SLA tier, support model and observability	Production readiness %
Adoption evidence	Consumers, subscriptions, usage trends	Reuse and adoption rate

Product Mindset

The catalog is where APIs stop being hidden technical assets and become reusable business products.

9. Governance and Policy Enforcement

API governance is the discipline that keeps APIs consistent, secure, documented and responsibly managed. A strong API Catalog turns governance from a document into an operating control. The catalog records evidence and gives policy engines something to validate.

Governance should be pragmatic. The goal is not to slow delivery. The goal is to make good API delivery repeatable. The catalog provides the mechanism for design standards, ownership, security, documentation and lifecycle enforcement.

Governance Pillar	Catalog Evidence	Policy Question
Design standards	Style, naming, domain, versioning and contract	Does the API follow enterprise API design rules?
Ownership	Business owner, technical owner, support team	Who is accountable?
Security baseline	Auth method, scopes, TLS, rate limit, sensitivity	Is the API safe enough to publish?
Documentation completeness	Specification, docs, examples, errors, changelog	Can consumers use the API responsibly?
Lifecycle enforcement	Draft, review, published, deprecated, retired	Is the API appropriately visible and supported?

Governance Approach

Good governance is not a meeting. It is evidence, policy and automation stitched into the API lifecycle.

10. Security and Compliance Controls

API Catalog is a security instrument because security teams cannot protect APIs they cannot see. Catalog records should show ownership, exposure level, authentication approach, authorization scope, data sensitivity, compliance tags and operational evidence.

OWASP API Security Top 10 2023 includes API risks such as Broken Object Level Authorization, Broken Authentication, Unrestricted Resource Consumption, Security Misconfiguration and Improper Inventory Management. The catalog directly supports inventory management by making APIs visible, owned and governed.

Security Concern	Catalog Control	Measure
Unknown APIs	Discovery and mandatory registration	Shadow API count
Weak ownership	Business and technical owner fields	Owner completeness %
Weak authentication	Auth method and scope fields	APIs without accepted auth method
Sensitive data exposure	Data classification and compliance tags	Sensitive APIs with full controls
Poor lifecycle hygiene	Deprecation and retirement status	Zombie / deprecated API ratio
Audit readiness	Reviewer, review date, policy exception and evidence links	Audit evidence preparation effort

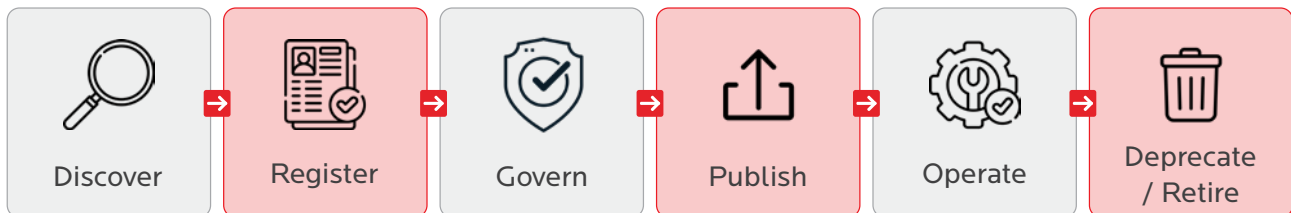
Security Message

API Catalog should be the API attack-surface register for security leadership.

11. Lifecycle Management

Every API has a lifecycle. It is designed, built, published, consumed, monitored, changed, deprecated and eventually retired. Without lifecycle management, organizations accumulate stale APIs that are still reachable, still consuming resources and sometimes still exposing risk.

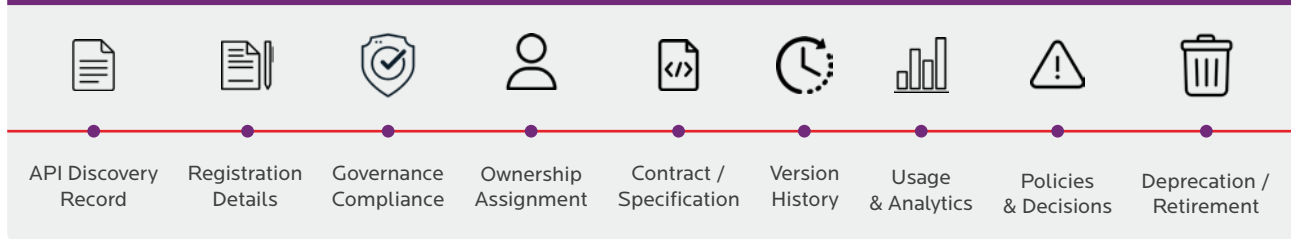
API Catalog Lifecycle Control Flow



Catalog evidence follows the API from initial discovery through retirement, keeping ownership, contract, policy and decision records traceable.



Catalog Evidence



☐ Active Lifecycle Stages

☐ Transition / Control Stages

Lifecycle Stage	Catalog Responsibility	Control Outcome
Discover	Find existing and unknown APIs	Baseline visibility
Register	Assign identity, owner and domain	Accountability
Govern	Validate standards, security and documentation	Quality assurance
Publish	Make API discoverable to approved audience	Controlled consumption
Operate	Track usage, incidents, SLA and consumers	Operational confidence
Deprecate	Communicate migration and sunset plan	Change safety
Retire	Archive evidence and remove exposure	Risk reduction

12. Comparison Table:

Catalog, Registry, Gateway, Portal

API Catalog is often confused with adjacent API capabilities. The distinction matters because replacing a catalog with a gateway list or a developer portal alone will not solve portfolio visibility and governance.

Capability	Primary Purpose	Optimizes	What It Does Not Fully Solve Alone
API Catalog	Governed discovery and portfolio visibility	Find, trust, reuse, govern and measure APIs	Runtime traffic control
API Registry	Technical registration of service definitions or endpoints	Technical reference and registration	Business context, governance and lifecycle portfolio
API Gateway	Runtime enforcement, routing and traffic control	Security policies, throttling, routing and transformation	Portfolio discovery and product metadata
Developer Portal	Consumer onboarding and documentation experience	Docs, try-it, subscriptions and support	Enterprise-wide governance by itself
API Marketplace	External API productization and monetization	Commercial discovery and subscription plans	Internal control of all APIs unless integrated with catalog

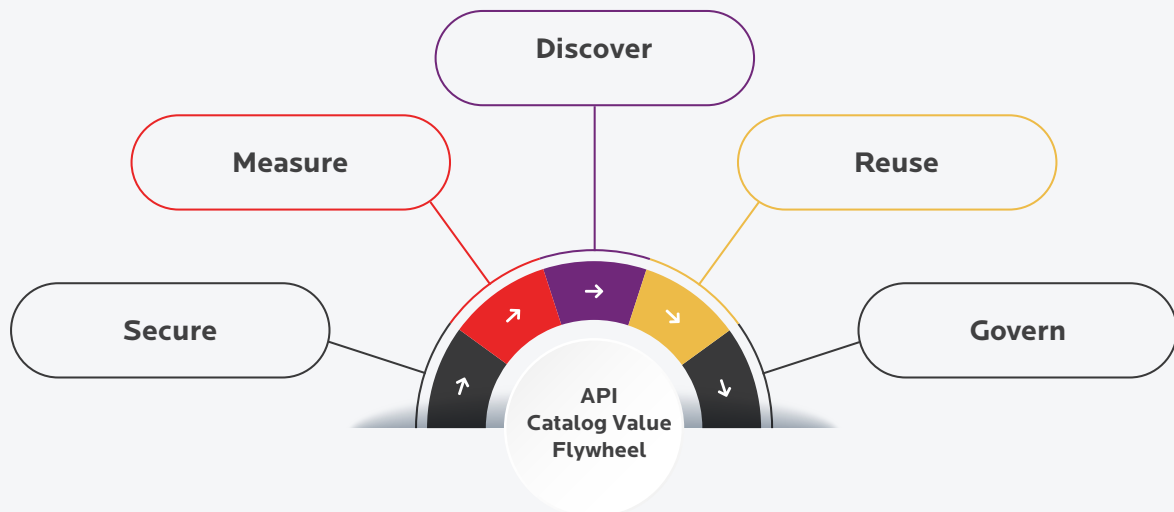
Clarification

A gateway runs API traffic. A catalog governs API knowledge. A portal improves API consumption. Mature enterprises need these capabilities to work together.

13. Benefits and Measures

The value of API Catalog should be expressed through measurable outcomes. The strongest executive story is not the number of APIs documented. The strongest story is what changes because APIs become discoverable, governed and reusable.

How API Catalog Compounds Value



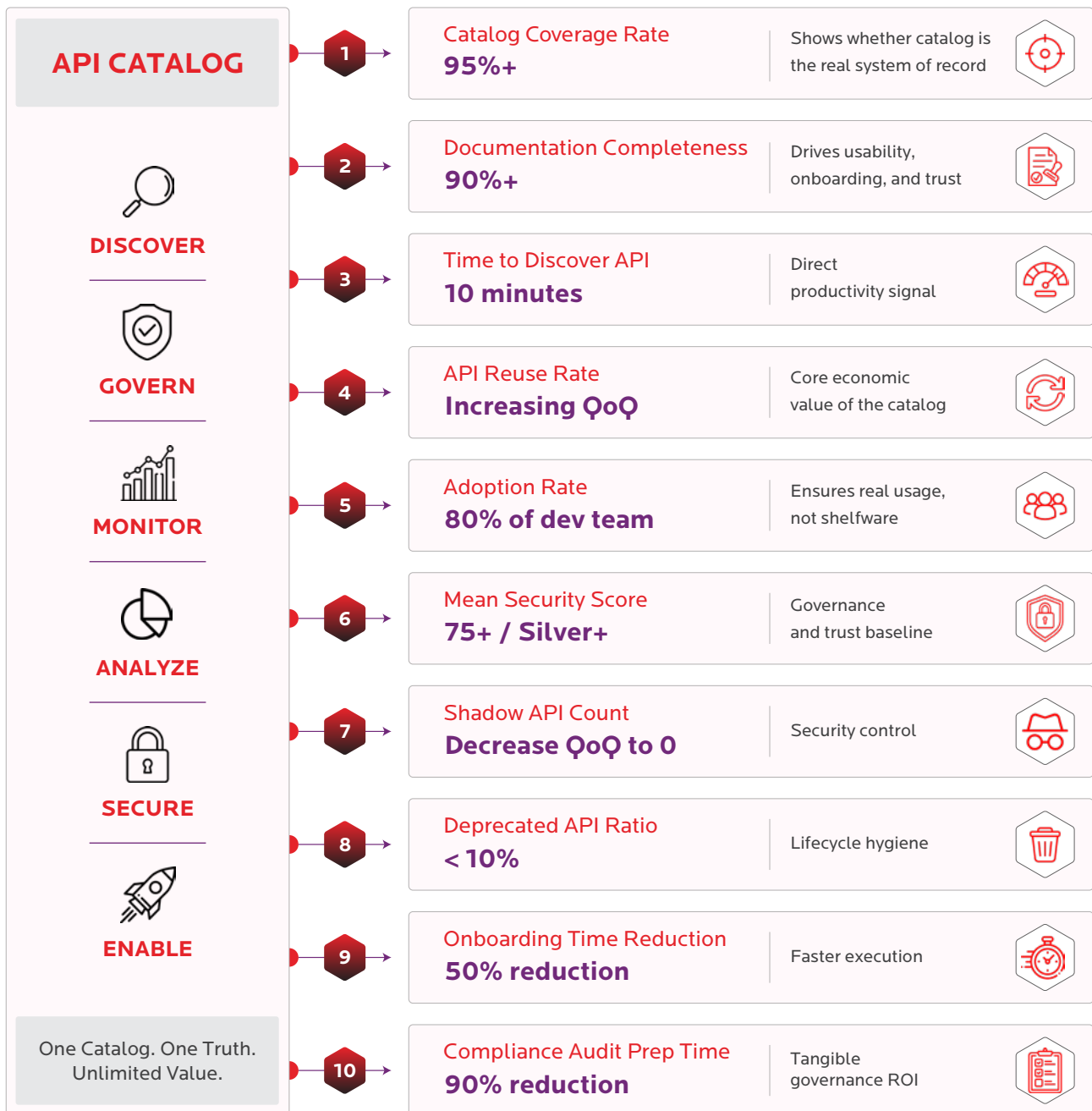
Benefit	Measurement	Target Direction
Faster discovery	Average time to find the right API	Reduce
Better reuse	% of new integrations using existing APIs	Increase
Less duplicate build	Duplicate APIs detected and avoided	Reduce
Improved security visibility	Shadow APIs and APIs missing owner/auth	Reduce
Audit readiness	Time to produce API inventory evidence	Reduce
Lifecycle hygiene	Deprecated or zombie APIs still active	Reduce
AI readiness	APIs with complete contract and intent metadata	Increase

14. ROI Model and Financial Measurement

A practical ROI model for API Catalog combines cost avoidance, productivity improvement, risk reduction and growth enablement. Not every outcome must be forced into a currency value, but every outcome should have a measurable signal.

API CATALOG - PORTFOLIO KPI DASHBOARD

Key Metrics That Drive Value, Efficiency, and Governance



API Catalog ROI Formula

Executive ROI Formula	ROI = avoided duplicate build cost + developer discovery time saved + audit/compliance effort reduced + lifecycle cost reduction + API reuse/productization value - catalog platform and operating cost.
----------------------------------	--

ROI Lever	Measurement Method	Financial Interpretation
Duplicate build avoidance	Avoided duplicate APIs × average build cost	Direct engineering cost avoidance
Discovery productivity	Hours saved per developer × developer count × loaded hourly rate	Productivity value
Audit effort reduction	Before/after hours to prepare inventory evidence	Compliance productivity
Security risk reduction	Decrease in shadow APIs and unmanaged exposure	Risk reduction indicator
Reuse value	Reuse count × avoided integration effort	Reusable asset value
Productization growth	External consumers, subscriptions, partner onboarding or API revenue	Growth enablement

15. CXO Metrics Dashboard

A CIO or CTO dashboard for API Catalog should show portfolio health, not only catalog activity. Recommended indicators should be reviewed monthly for operational teams and quarterly for executive governance. The dashboard must make progress visible and trigger action when controls are weak.

Metric	Meaning	Executive Interpretation
Catalog coverage rate	Percent of known production APIs represented in catalog	Shows visibility maturity
Owner completeness	Percent of APIs with business and technical owner	Shows accountability
Contract coverage	Percent of APIs with machine-readable contract link	Shows automation and documentation readiness
Documentation completeness	Percent of APIs meeting documentation minimum	Shows consumer readiness
Reuse rate	Percent of demand fulfilled by existing APIs	Shows productivity and portfolio leverage
Shadow API count	Unregistered APIs discovered	Shows risk exposure
Deprecated /zombie ratio	APIs marked deprecated or stale but still active	Shows lifecycle hygiene
AI readiness score	APIs with contract, intent, security and docs suitable for machine consumers	Shows future readiness

Dashboard Principle

If a metric does not influence a decision, it is only reporting. API Catalog KPIs should drive investment, remediation and reuse behavior.

16. AI Readiness and Machine-Consumable APIs

API Catalog is becoming more important because APIs are increasingly consumed by tools, automations and AI agents. Human-readable documentation remains important, but machine consumers need structured contracts, reliable intent, security constraints and policy context.

Postman's 2025 State of the API Report says APIs are no longer just powering applications; they are powering agents. The report also states that it surveyed more than 5,700 developers, architects and executives and that API strategy is becoming AI strategy.

AI Need	Catalog Requirement	Measurement
Discover APIs	Search, tags, RFC 9727 discovery surface	Discoverable APIs %
Understand intent	Business description, examples and semantic tags	Intent metadata completeness
Understand contract	OpenAPI, AsyncAPI or other machine-readable specification	Contract coverage %
Invoke safely	Auth method, scopes, rate limits, sensitivity and lifecycle state	Security readiness %
Trust quality	Owner, quality score, documentation and lifecycle evidence	AI readiness score

AI-era Message

AI agents will not ask a colleague where an API is. They need a trusted catalog that can tell them what exists, what it does and whether it is safe to use.

17. Implementation Roadmap

A successful API Catalog program should start small, prove value and then scale. The early goal is not to catalog every API with poor quality. The early goal is to catalog the most critical APIs with high-quality metadata and visible adoption benefits.

Phase	Objective	Key Deliverables	Success Measure
Discover	Create baseline inventory	Gateway import, repository scan, team inputs, shadow list	Initial coverage rate
Define	Agree metadata and governance model	Gateway import, repository scan, team inputs, shadow list	Approved operating model
Populate	Register priority APIs	Owners, specs, docs, sensitivity, SLA	Useful searchable catalog
Enable	Drive consumer adoption	Search, access workflow, onboarding	Active users and search success
Enforce	Embed in delivery lifecycle	CI/CD gates, security checks, policy workflow	Active users and search success
Optimize	Improve and extend	Dashboards, deprecation, AI readiness, partner catalog	Quarterly KPI improvement

First 90-Day Focus

- Prioritize the top business-critical and most-consumed APIs.
- Define minimum metadata: name, domain, owner, contract, authentication, lifecycle, documentation, SLA and data sensitivity.
- Introduce Gold/Silver/Bronze quality scoring.
- Make “search before build” part of integration demand intake.
- Publish executive metrics: coverage, reuse, shadow APIs, zombie APIs and AI readiness.

18. Operating Model and Roles

API Catalog success depends on clear roles. A catalog without operating accountability becomes a document store. A catalog with accountable roles becomes a living platform capability.

Role	Primary Accountability	Catalog Interaction
CXO / Steering Group	Strategic funding, outcome tracking and risk prioritization	Reviews KPI dashboard and investment decisions
Enterprise Architect	Business capability alignment and standards	Defines catalog taxonomy and design governance
API Product Owner	API business purpose, roadmap and consumer value	Maintains product context and adoption goals
Technical Owner	Implementation, contract and runtime readiness	Maintains contract, docs, lifecycle and support data
Security / Compliance	Risk controls, audit evidence and policy exceptions	Reviews sensitivity, controls and exceptions
Platform Team	Catalog platform, integrations and automation	Operates catalog workflows and CI/CD integration
Developer / Consumer	Find, evaluate and use APIs	Searches catalog, requests access and provides feedback

Operating Model Principle

Operating model is what keeps the catalog alive after launch. Without ownership hygiene, metadata quality will decay.

19. CXO Decision Framework and Conclusion

A CXO should evaluate API Catalog as an enterprise capability, not a tooling feature. The decision should be based on whether the organization can see, trust, reuse, govern and measure APIs across the enterprise.

Decision Area	Question for Leadership	Recommended Position
Strategy	Are APIs managed as reusable business assets?	Adopt API product ownership and portfolio KPIs
Risk	Can security list API exposure and ownership quickly?	Make catalog the API attack-surface register
Delivery	Do teams search before building new integrations?	Make catalog discovery part of demand intake
Governance	Are policies automated or manually checked late?	Embed catalog controls into CI/CD
AI readiness	Can machine consumers discover safe APIs?	Add contract, intent and security metadata
Value	Can reuse and avoid waste be measured?	Track benefits quarterly

API Catalog is foundational for digital maturity. It improves visibility, discovery, reuse, governance, security, lifecycle discipline, and AI readiness. The organizations that build disciplined API Catalog capabilities now will have a stronger foundation for ecosystem integration and AI-era automation.

Final Takeaway

The API Catalog should become the trusted enterprise map of digital capabilities - where APIs are discovered, governed, measured and prepared for AI-era consumption.

20. Annexure — References and Links

The white paper uses the uploaded API Catalog master paper as the primary internal source and complements it with public global standards and industry sources below.

Ref	Source	Link
A1	RFC 9727 - api-catalog: A Well-Known URI and Link Relation to Help Discovery of APIs	https://www.rfc-editor.org/rfc/rfc9727.html
A2	RFC 9727 PDF - RFC Editor	https://www.rfc-editor.org/rfc/rfc9727.pdf
A3	OpenAPI Initiative - OpenAPI Specification	https://www.openapis.org/
A4	OpenAPI Specification Repository	https://github.com/OAI/OpenAPI-Specification
A5	AsyncAPI Initiative - Event-driven API specification	https://www.asyncapi.com/
A6	AsyncAPI Specification Repository	https://github.com/asyncapi/spec
A7	Postman 2025 State of the API Report	https://www.postman.com/state-of-api/2025/
A8	OWASP API Security Top 10 2023	https://owasp.org/API-Security/editions/2023/en/0x00-header/
A9	MuleSoft - Forrester TEI study finds 426% ROI from MuleSoft	https://www.mulesoft.com/lp/reports/integration-total-economic-impact

Authors



Nitin Gajale

Associate VP, Enterprise Integration and Automation Practice, Birlasoft



Prakash Umarania

Lead Architect,
Enterprise Integration Practice

Accelerate **Impact**

Birlasoft combines the power of domain, enterprise, and digital technologies to reimagine business processes for customers and their ecosystem. Its consultative and design-thinking approach makes societies more productive by helping customers run businesses. As part of the multibillion-dollar diversified CKA Birla Group, Birlasoft with its 12,000+ professionals, is committed to continuing the Group's 170-year heritage of building sustainable communities.

Copyright © 2026. Birlasoft and its logo(s) are trademarks of Birlasoft All rights reserved.
All other logo(s) used are trademarks of their respective owners.

contactus@birlasoft.com | birlasoft.com



RESOURCES