

Das API-Benutzerhandbuch

7 Best Practices erfolgreicher API-Teams

Manfred Bortenschlager

Director, Business Development for API-based
Integration Solutions and API Management,
Red Hat

Andrew Mackenzie

Director of Software Engineering API
Management, Red Hat

Jaya Baskaran

Principal Technical Marketing Manager,
Red Hat

Greg Pack

Senior Product Marketing Manager
App Services and API Management Solutions,
Red Hat



Inhalt

Zusammenfassung	3
Einleitung	3
Aus welchem Grund möchten wir APIs implementieren?	4
Wer wird diese APIs voraussichtlich nutzen?	4
Welche konkreten Ergebnisse möchten wir damit erzielen?	4
Wie planen wir die API-Ausführung, um diese Ziele zu erreichen?	5
Das API-Team	5
Best Practice Nr. 1: Starker Fokus auf den Wert der API	6
Was bedeutet das in Bezug auf die API?	6
Überlegungen zum API-Programmwert	7
Best Practice Nr. 2: Von Anfang an ein klares Geschäftsmodell	7
Ein API-Geschäftsmodell verstehen	7
Überlegungen zu Ihrem API-Geschäftsmodell	8
Best Practice Nr. 3: Design und Implementierung auf die Nutzenden abstimmen	9
Einfachheit.....	9
Flexibilität.....	9
Überlegungen zur Gestaltung des API-Protokolls	9
Integration mit Produkten	9
Infrastruktur für API-Implementierung und -Deployment	10
Best Practice Nr. 4: Machen Sie API-Operationen zur Priorität	10
API-Operations-Donut.....	10
Wesentliche Komponenten der API-Verwaltung	11
Überlegungen zu API-Operationen.....	11
Best Practice Nr. 5: Das Entwicklererlebnis ist der Schlüssel	12
API-Erfolg ist mehr als tolles Design	12
Überlegungen zur Evaluation des Entwicklererlebnisses	13
Best Practice Nr. 6: Mehr als nur das 1x1 des Marketings	13
Überlegungen zum API-Marketing	13
Best Practice Nr. 7: Denken Sie an Abschaltung und Änderungsmanagement von APIs	14
Überlegungen zur Lebensdauer von APIs	14
Nachhaltigkeit Ihrer API-Strategie.....	15
Erstellen eines langlebigen API-Programms.....	15
APIs existieren nicht in einem Vakuum	15
Warum sollten Sie Red Hat für die Entwicklung und Verwaltung Ihrer APIs nutzen?	16
Fazit	16

Zusammenfassung

Fortschritte in der modernen Anwendungsentwicklung, Konnektivität und Infrastruktur wie 5G und Edge Computing verändern weiterhin, wie und wo Geschäftsabläufe stattfinden. Diese Fortschritte beschleunigen zwar die Geschäftsabläufe, aber ohne moderne APIs (Application Programming Interfaces) wäre dies nicht möglich.

APIs sind heute zur digitalen Verbindung moderner Unternehmen geworden und bieten neue Möglichkeiten für verschiedene Bereiche, von Abläufen über Produkte bis hin zu Partnerschaftsstrategien. Unabhängig davon, ob Sie bereits über eine API-Strategie verfügen oder diese vorbereiten, finden Sie und Ihr Team in diesem E-Book die 7 Best Practices für erfolgreiche API-Programme.

Einleitung

Bevor Sie dieses E-Book lesen, sollten Sie über die wichtigsten Ziele Ihres API-Programms nachdenken. Einige der späteren Fragen können dazu beitragen, dass Sie Ihre Ziele ändern, bestätigen oder ihnen mehr Bedeutung verleihen.

Ein effektives API-Programm sollte auf der übergeordneten Unternehmensstrategie aufsetzen und zur Erfüllung ihrer Ziele beitragen. Sie wissen, dass Sie es mit einer tollen Strategie zu tun haben, wenn Sie die folgenden 4 Fragen beantworten können:

1. Aus welchem Grund möchten wir APIs implementieren?
2. Wer wird diese APIs voraussichtlich nutzen?
3. Welche konkreten Ergebnisse möchten wir damit erzielen?
4. Wie planen wir die API-Ausführung, um diese Ziele zu erreichen?

Aus welchem Grund möchten wir APIs implementieren?

Bei der Betrachtung von APIs ist ein umfassendes Verständnis ihres Wertes und Zwecks unerlässlich. Zur Maximierung ihres potenziellen geschäftlichen Nutzens müssen Fehlannahmen aufgedeckt werden. Laut einem weit verbreiteten Irrglauben sind APIs nur dann wertvoll, wenn die Nutzer dafür bezahlen. Dies trifft zu, wenn die API das Produkt ist. Viele API-Projekte bleiben jedoch intern und generieren verschiedene Metriken wie verkürzte Markteinführungszeiten, Wiederverwendbarkeit, Umsatz, Markenbekanntheit oder Partnerempfehlungen.

Fünf typische Use Cases für API-Anbieter sind:

1. **Skalieren des Multichannels.** Ausweiten der Zugänglichkeit über verschiedene Kanäle wie etwa mobile Apps und das IoT (Internet of Things).
2. **Vergrößern Ihres Partnernetzwerks.** Stärken von Kunden- oder Partnernetzwerken für die Ausweitung.
3. **Vergrößern Ihrer Reichweite.** Aufbau umfassender Netzwerke für Transaktionen oder Inhaltsdistribution.
4. **Fördern neuer Geschäftsmodelle.** Verbesserte Innovation durch neue Geschäftsmodelle.
5. **Schaffen interner Innovationen.** Schaffen von Innovationen innerhalb des Unternehmens.

Viele Beispiele für erfolgreiche API-Strategien sind in modernen Technologieunternehmen zu finden, darunter Amazon, Salesforce, Twilio und weitere. Das heißt jedoch nicht, dass nur diese führenden Technologieunternehmen APIs für Monetarisierung, Innovation, Stärkung von Partnerschaften oder Agilität nutzen können. Viele gut etablierte Unternehmen verfügen über zahlreiche traditionelle Systeme, die nicht plötzlich und wahrscheinlich sogar nie verschwinden werden. APIs sind der perfekte Mechanismus, um diese herkömmlichen Systeme zu integrieren, sie intern oder extern zugänglich zu machen und sie zum Schaffen neuer Werte mit anderen Diensten zu verknüpfen. Unternehmen, die sich auf dem Weg der Digitalisierung befinden, können eine gut strukturierte API-Strategie implementieren und so die Vorteile der API Economy nutzen.

Diese Motivation muss so stark sein, dass die Entscheidung zur Investition in APIs für das Unternehmen eine offensichtliche Wahl ist.

Wer wird diese APIs voraussichtlich nutzen?

Wenn Sie wissen, wer Ihre API-Endnutzende sind, können Sie die Erfolgsmetriken Ihres API-Programms besser definieren.

- **Interne Endnutzende.** APIs können von internen Teams zum Zugriff auf Unternehmensdaten für eine Vielzahl von Geschäftszwecken genutzt werden.
- **Externe Endnutzende.** Die gleichen APIs oder ein Teil davon können von externen Drittentwicklern in ihren Anwendungen verwendet werden.

Diese Motivation muss so stark sein, dass die Entscheidung zur Investition in APIs für das Unternehmen eine offensichtliche Wahl ist.

Welche konkreten Ergebnisse möchten wir damit erzielen?

Zur Ermittlung der durch APIs erzielten konkreten Ergebnisse sollten Sie interne und externe organisatorische Aspekte berücksichtigen:

- **Interne Aspekte.** Nutzen Sie Ihre besonderen und wertvollen Assets. Organisationen, die über besondere Daten verfügen, wie beispielsweise die „Gefällt mir“-Daten von Meta, können wertvolle APIs anbieten.
- **Externe Aspekte.** Beachten Sie Marktdynamik, Trends, Mitbewerber und Kundenverhalten. Externe Faktoren prägen Unternehmensstrategien und beeinflussen API-Funktionen.

Die Marktdynamik wirkt sich stark auf Mapping-APIs aus. Frühe Mapping-Unternehmen haben die Echtzeit-Nachfrage übersehen und damit Startups wie Waze zum Erfolg verholfen. Google übernahm Waze und integrierte dessen Technologie in eine erfolgreiche API. Auch Twitter, Reddit und andere Größen haben APIs für Innovationen und das Wachstum ihres Netzwerks genutzt.

Die richtige API-Strategie kombiniert häufig interne Assets mit externen Markterkenntnissen.

Wie planen wir die API-Ausführung, um diese Ziele zu erreichen?

Bei der abschließenden Frage „Wie muss unser API-Programm aussehen, damit es unsere Ziele erfüllt?“ geht es um Implementierung und Ausführung, was eine sorgfältige Planung der folgenden Punkte erfordert:

1. **Technologie-Auswahl.** Bestimmen Sie den Technologie-Stack für die Erstellung von APIs.
2. **Design und Wartung.** Ausarbeitung von API-Design- und Wartungsstrategien.
3. **Werbung und Marketing.** Förderung von APIs innerhalb der Organisation und deren Vermarktung nach außen.
4. **Zuweisung von Ressourcen.** Zuteilung der notwendigen Ressourcen für Entwicklung und Wartung der API.
5. **Zusammensetzung des Teams.** Stellen Sie ein kompetentes Team für Entwicklung und Verwaltung der API zusammen.
6. **Entwicklungs-Community.** Erstellung und Verwaltung eines speziellen Kommunikationsplans für externe und interne Entwickler.
7. **Erfolgsmetriken.** Festlegung von Methoden zur Erfolgskontrolle in Bezug auf die Unternehmensziele.

Indem sie das „Warum“, „Wer“, „Was“ und „Wie“ von APIs aufdecken, können Unternehmen Innovation und Wachstum in der sich entwickelnden API Economy voranbringen. Die Antworten auf diese Fragen sind für jedes Unternehmen unterschiedlich und werden von Ihren Zielen, der von Ihnen verfolgten Strategie und insbesondere dem API-Team beeinflusst.

Das API-Team

Ein API-Team ist normalerweise so wie andere Produktteams strukturiert. Egal ob Sie es mit internen oder externen Kunden zu tun haben: Das API-Team ist dafür verantwortlich, die Infrastruktur zu entwickeln, zu implementieren, zu optimieren und auszuführen, auf die sich andere verlassen.

Genau wie Produktteams können auch API-Teams vielfältig sein. Die Teams sollten folgende Mitarbeitende beinhalten: eine produktorientierte Person, die Strategie und Ziele überwacht; designorientierte Mitglieder, welche die Best Practices beim API-Design umsetzen; Engineers, die die API-Technologie programmieren, und Operations-Mitglieder, die letztlich die API ausführen. Im Laufe der Zeit können weitere Mitarbeitende hinzukommen, beispielsweise Mitglieder der Support- und Community-Teams sowie API- und Sicherheitsfachleute. Zu Ihrem erweiterten API-Team könnten auch Mitglieder Ihrer Entwicklungs-Community gehören.

Auch wenn dieses Team aus zahlreichen Personen bestehen kann, haben manche davon mehrere Aufgaben zu erfüllen, vor allem in kleineren Unternehmen. Es muss sichergestellt werden, dass die Meinungen aller Stakeholder vertreten sind und mindestens ein Teammitglied nach deren Bedenken fragt.

In vielen Fällen bestehen API-Teams nur vorübergehend und gehören zu verschiedenen Organisationseinheiten mit unterschiedlichen Vorgesetzten. Diese Struktur kann die Definition einer gemeinsamen Vision für die API erheblich erschweren. Bei großen API-Programmen müssen unter Umständen verschiedene API-Teams zusammenarbeiten.

Unabhängig von der Größe Ihres Unternehmens werden die 7 in diesem E-Book beschriebenen Best Practices dazu beitragen, ein erfolgreiches API-Team aufzubauen, das vielleicht mehr Personen umfassen wird, als Sie denken.

Best Practice Nr. 1: Starker Fokus auf den Wert der API

Bei API-Programmen muss das Hauptziel in der Wertschöpfung sowie der Vermeidung von Komplexität liegen. Das Leistungsversprechen bestimmt den Wert für Nutzende, ein entscheidender Faktor für den Erfolg einer API. Ein überzeugendes Leistungsversprechen ist für wirksames Marketing unerlässlich. Dieses mit den Unternehmenszielen abzustimmen gewährleistet Nachhaltigkeit und ermöglicht es etablierten Unternehmen, ihr Angebot durch APIs zu erweitern.

Das Modell des API-Werts von Alex Osterwalder bringt die Vorteile für die Nutzenden mit den API-Funktionen in Einklang und schafft so ein entscheidendes Zusammenspiel zwischen den Bedürfnissen Ihrer Nutzenden und dem API-Wert.¹ Es ist entscheidend, auf Bedürfnisse einzugehen, Probleme zu beheben und Mehrwert zu schaffen.

Was bedeutet das in Bezug auf die API?

In diesem iterativen Prozess beschreibt der 1. Schritt die Aufgaben, die Ihre Nutzenden zu erledigen versuchen, wie etwa das automatische Versenden dringender Mitteilungen in Notfällen, das Sichern wichtiger Dateien oder das Sammeln von Daten zur Erkennung bestimmter Events.

Im 2. Schritt ermitteln Sie die speziellen Probleme, denen sich die Nutzenden vor, während oder nach der Durchführung einer Aufgabe gegenübersehen. Dazu gehören der zuverlässige Versand von Nachrichten mit mehreren Versuchen, Fehlererkennung, Verwaltung mehrerer Nachrichten, Integration standortbasierter Nachrichtensysteme, Gewährleistung der Dateizustellung bei minimaler Bandbreitennutzung und Echtzeitkorrelation umfangreicher Datenmengen.

Der 3. Schritt beim Erstellen eines Benutzerprofils umfasst die Darstellung potenzieller Vorteile, wie etwa andere Arten von Benachrichtigungen, die Möglichkeiten schaffen anstatt vor Gefahren zu warnen, die Eliminierung anderer Storage-Geräte, wenn die Zuverlässigkeit gut genug ist, und das automatische Auslösen von eventbasierten Aktionen.

Auf der Value Map sollten Funktionalität, Features und Dienste der API dargestellt werden, wobei der Schwerpunkt darauf liegen sollte, Probleme zu lösen und Gewinne zu erzielen. Der Prozess mündet in konkreten Beispielen wie einer Messaging-API, die die Zustellung von Nachrichten sicherstellt, einer Storage-Synchronisations-API für effiziente Versionsaktualisierungen und einer Datenaggregations-API, die konfigurierbare Datenströme bereitstellt.

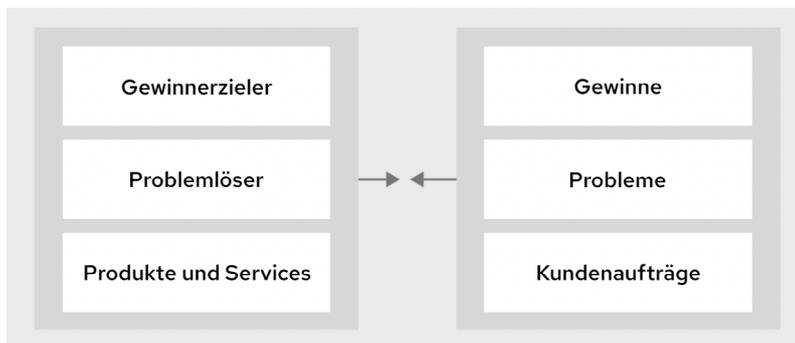


Abbildung 1. Übersicht der Leistungsversprechen

Schließlich sollte das API-Team im Rahmen einer Übung mehrere Aussagen treffen, die konkret das Zusammenspiel zwischen API und Nutzerprofil aufzeigen. Wenn Sie Ihre Aussagen über das Zusammenspiel von API und Nutzerprofil auf eine übergeordnete Aussage reduzieren können, wird diese zum Leistungsversprechen für Ihre APIs. Diese Aussagen demonstrieren die Ausrichtung der API auf die Nutzenden und präzisieren das Leistungsversprechen. Im Falle der Messaging-API kann das folgendermaßen aussehen:

„Das Kundenzitat steht hier. Achten Sie beim Hinzufügen von Seitenleisteninhalten auf die Ausdehnung der Sprache. Einige Sprachen, wie Deutsch, benötigen mehr Platz als Englisch. Lassen Sie mindestens 1/4 der Seitenleiste für die Übersetzung frei.“

Sie denken vielleicht: „Das scheint mir total überzogen, wir haben doch nur eine interne API“. Das ist eine ganz normale Reaktion. Der Fokus auf den Wert ist jedoch auch für interne Use Cases äußerst wichtig. Ist das Leistungsversprechen nicht klar definiert, wird es sehr viel Zeit erfordern, die API anderen Teams vorzustellen und an diese zu vermitteln. Mit einer wohlgedachten Formulierung hingegen wird das API-Programm ein wertvoller Beitrag zum Unternehmen.

¹ Strategizer, „[Value Proposition Canvas](#).“ Zugriff im September 2023.

Überlegungen zum API-Programmwert

Berücksichtigen Sie diese wichtigen Punkte bei der Definition des Werts Ihres API-Programms:

- 1. Identifizierung der Nutzenden.** Identifizieren Sie Nutzende auf Grundlage ihrer Beziehungen (Kunden, Partner und Entwickler), Rollen und Vorlieben.
- 2. Probleme und Vorteile ansprechen.** Anhand des in Abbildung 1 dargestellten Leistungsversprechens werden die Probleme, Vorteile und wichtigen Bedürfnisse der Nutzenden ermittelt. Messen Sie Verbesserungen der Metriken (Geschwindigkeit, Umsatz und Kosten) und das Potenzial für neue Möglichkeiten.
- 3. Unterstützte Use Cases.** Nutzen Sie das Leistungsversprechen, um wirksame Problemlösungen zu identifizieren und Einnahmequellen zu erschließen. Konzipieren Sie Ihre API so, dass sie für diese spezifischen Use Cases geeignet ist.
- 4. Zukünftige Wertsteigerung.** Planen Sie Ihr Leistungsversprechen mit Blick auf die Zukunft. Antizipieren Sie kommende Meilensteine, Trends oder technologische Innovationen, um eine kontinuierliche Wertsteigerung zu erreichen.
- 5. Interner organisatorischer Wert.** Bewerten Sie den internen Nutzen der API und den potenziellen Wert für andere Teams.

Durch die Beantwortung dieser Fragen können Sie für Ihr API-Programm ein klares Leistungsversprechen abgeben und sicherstellen, dass es auf die Bedürfnisse der Nutzenden und die Unternehmensziele abgestimmt ist.

Best Practice Nr. 2: Von Anfang an ein klares Geschäftsmodell

Die Entwicklung einer erfolgreichen API geht über ein reines Leistungsversprechen hinaus. Dazu muss die API an ein klar definiertes Geschäftsmodell gekoppelt werden. Es ist zwar wichtig, den Wert der API zu erkennen und zu vermitteln, aber die Kosten müssen gegen den finanziellen oder greifbaren Nutzen abgewogen werden. In seinem Buch „Business Model Generation“, das über 400 Mitwirkende hat, definiert Alex Osterwalder das Geschäftsmodell eines Unternehmens als die Art und Weise, wie das Unternehmen Werte vorschlägt, schafft, liefert und erfasst.

Ein API-Geschäftsmodell verstehen

In der Übersicht zum Geschäftsmodell werden die Kernkomponenten eines Geschäftsmodells aufgeschlüsselt, darunter:²

- 1. Leistungsversprechen.** Definiert den besonderen Wert der API.
- 2. Umsatzquellen.** Beschreibt, wie die API Einnahmen generiert.
- 3. Kostenstruktur.** Kosten für die Wartung der API.
- 4. Kundensegmente.** Nutzenden-Zielgruppen.
- 5. Kundenbeziehungen.** Einbindung der Nutzenden durch das Unternehmen.
- 6. Kanäle.** Distributionsmethoden zum Erreichen der Nutzenden.
- 7. Wichtige Partner.** Für den Erfolg der API entscheidende Kooperationen.
- 8. Wichtige Tätigkeiten.** Für den Betrieb der API erforderliche Aufgaben.
- 9. Wichtige Ressourcen.** Für die API-Implementierung notwendige, wesentliche Assets.



Abbildung 2. Übersicht zum Geschäftsmodell

APIs können unter der Verwendung vorhandener Assets neue Möglichkeiten eröffnen. Es ist zwar entscheidend, den Wert der API zu kennen, aber dennoch müssen auch ihre Kosten kontrolliert werden. Ein ungeeignetes Modell kann die Kosten trotz des API-Werts in die Höhe treiben.

Überlegungen zu Ihrem API-Geschäftsmodell

Bringen Sie Ihr Geschäftsmodell mit der API-Nutzung in Einklang und beachten Sie dabei diese 5 wichtigen Bereiche:

- 1. API-Wert für die Organisation.** Untersuchen Sie, wie die API der Organisation hilft, die Reichweite zu vergrößern oder Innovationen voranzutreiben, um so verschiedene und nicht nur monetäre Werte zu bewerten.
- 2. Erfassen von Werten.** Bestimmen Sie den besten Weg, um den identifizierten Wert zu erfassen, und minimieren Sie die Hindernisse für eine maximale Wertrealisierung.
- 3. Kostendeckung.** Erkennen Sie API-bezogene Kosten, die oft über das API-Team hinausgehen, wie Technik oder Marketing, und planen Sie deren Deckung ein.
- 4. Langfristige Verpflichtung.** Seien Sie sich bewusst, dass APIs über die Anfangsinvestition hinaus ein kontinuierliches Engagement für Betrieb und Wartung erfordern.
- 5. Strategische Partnerschaften.** Identifizieren Sie wichtige Partnerschaften und nutzen Sie ergänzende Angebote von Partnern und Zulieferern während der Entwicklung und Markteinführung der API.

Best Practice Nr. 3: Design und Implementierung auf die Nutzenden abstimmen

Das API-Design umfasst grundlegende Designprinzipien für ein konsistentes Benutzererlebnis. Ziel ist es, ein einsatzbereites Design zu erreichen, das erfahrene Entwicklungsteams APIs intuitiv verstehen. Das API-Design sollte sich auf 2 Kernbereiche konzentrieren: Einfachheit und Flexibilität.

Einfachheit

Die Einfachheit des API-Designs ist kontextabhängig, wobei die Komplexität der Designs je nach Use Case variiert. Eine ausgewogene Granularität der API-Methoden ist unerlässlich. Einfachheit lässt sich auf verschiedenen Ebenen betrachten:

1. **Datenformat.** Entscheiden Sie sich für XML, JSON, proprietäre Formate oder Kombinationen.
2. **Struktur der Methode.** Die Methoden reichen von allgemeinen bis hin zu sehr spezifischen Methoden, die oft sequenziert werden, um sie bestimmten Use Cases anzupassen.
3. **Datenmodell.** Das exponierte API-Datenmodell kann sich von dem zugrunde liegenden unterscheiden, was die Nutzbarkeit und Wartungsfreundlichkeit beeinflusst.
4. **Authentifizierung.** Verschiedene Authentifizierungsmechanismen haben Stärken und Schwächen, wobei die Wahl vom jeweiligen Kontext abhängt.
5. **Richtlinien für die Nutzung.** Rechte und Quotas für Entwicklungsteams sollten verständlich und überschaubar sein.

Flexibilität

Das Gleichgewicht zwischen Einfachheit und Flexibilität ist entscheidend. Eine zu vereinfachte API ist möglicherweise nur für bestimmte Use Cases geeignet, was ihre Anpassungsfähigkeit einschränkt. Um Flexibilität zu schaffen, definieren Sie den möglichen Einsatzbereich, einschließlich zugrunde liegender Systeme und Datenmodelle. Definieren Sie eine sinnvolle und nutzbringende Teilmenge der Operationen in diesem Einsatzbereich. Um die richtige Balance zwischen Einfachheit und Flexibilität zu finden, sollten Sie:

1. **Atomare Operationen offenlegen.** Kombinieren Sie atomare Operationen zur Abdeckung des gesamten Operationsbereichs.
2. **Häufige und wertvolle Use Cases identifizieren.** Entwerfen Sie eine zweite Schicht von Meta-Operationen, die atomare Operationen für diese Use Cases vereinigen.

Überlegungen zur Gestaltung des API-Protokolls

Auch wenn die REST-Architektur (Representational State Transfer) weiterhin der vorherrschende Standard für die API-Entwicklung ist, werden die API-Protokolle immer vielfältiger. Zu den neueren Konzepten gehören Streaming-APIs oder WebSockets. Auch die klassischen Webservice-APIs werden nicht verschwinden. Die Wahl des Protokolls sollte nach diesem Prinzip erfolgen: Wählen Sie das für Nutzende relevanteste Protokoll aus und finden Sie dann das richtige Gleichgewicht zwischen Einfachheit und Flexibilität. Die Infrastruktur zur Bereitstellung von REST-APIs entwickelt sich stetig weiter.

Berücksichtigen Sie bei Ihren Überlegungen zum API-Design folgende Bereiche:

1. **Ausrichtung auf Use Cases.** Entwickeln Sie die API zur Unterstützung identifizierter Use Cases und bewahren Sie dabei Flexibilität für innovative und seltenere Szenarien.
2. **Durchdachte RESTfulness.** RESTful-APIs sind zwar hochmodern, dennoch sollten Sie prüfen, ob diese wirklich Ihren Anforderungen entsprechen, da andere Architekturstile besser für bestimmte Use Cases geeignet sein könnten.
3. **Abstrahiertes Datenmodell.** Implementieren Sie eine API mit einer Abstraktionsschicht zwischen der API und dem Datenmodell, sodass ein direkter Datenbankzugriff nur bei Bedarf erfolgen kann.
4. **Geografische Überlegungen.** Berücksichtigen Sie nichtfunktionale Aspekte wie Latenzzeiten und Verfügbarkeit, indem Sie Rechenzentren strategisch in der Nähe Ihrer wichtigsten Nutzendenregionen platzieren.
5. **Integration mit Produkten.** Stellen Sie sicher, dass das API-Design durch Koordination oder Entkopplung mit anderen Produkten harmonisiert und gleichzeitig eine klare interne und externe Kommunikation der Struktur gewährleistet wird.

Infrastruktur für API-Implementierung und -Deployment

Mit dem Umstieg von Unternehmen auf Cloud- und Hybrid Cloud-Infrastrukturen für Daten und Anwendungen werden auch die Infrastrukturen für API-Implementierung und -Deployment vielfältiger. APIs können auf der Grundlage von Microservices erstellt und mit monolithischen oder anders aufgebauten APIs kombiniert werden. Sie können sich in Containern, virtuellen Maschinen (VMs), Bare Metal- oder in einer Public Cloud-Umgebung befinden.

Wie bei anderen Anwendungen auch ist eine automatisierte CI/CD-Pipeline (Continuous Integration and Continuous Delivery) für die Verwaltung des API-Lifecycles entscheidend. [Die Einführung von GitOps](#) und Argo CD kann ein zentralisiertes API-Konfigurationsmanagement und automatisierte Deployment-CD ermöglichen; außerdem wird die Zusammenarbeit im API-Team gefördert, was eine schnellere Entwicklung und bessere API-Produktqualität begünstigt.

Best Practice Nr. 4: Machen Sie API-Operationen zur Priorität

Das API-Plattformteam verwaltet die APIs, sobald sie in Betrieb sind und stellt sicher, dass sie zugänglich sind und gemäß den Erwartungen des Entwicklungsteams bereitgestellt werden. Auch wenn Anbieter über Lösungen verfügen, ist die Wahl der richtigen Strategie entscheidend für den Erfolg. Die API-Verwaltung umfasst 2 Hauptfunktionen:

1. Optimierung interner Prozesse zur Effizienzsteigerung und Kostensenkung.
2. Effektive Gestaltung der Abläufe, um die Erwartungen externer Programmentwicklungsteams zu erfüllen.

[Großartige APIs entwickeln Teil 1: Der Goldstandard](#) und der API Operations Donut in Abbildung 3 helfen Ihnen dabei, diese Ziele zu erreichen.

API Operations Donut

Der API Operations Donut kann zur Definition von Operationstaktiken verwendet werden, um die API-Strategie einer Organisation umzusetzen. Der innere Kreis des Donuts steht für die internen Aktivitäten und Auswirkungen einer Organisation, außerhalb des Rings werden die externen Auswirkungen dargestellt.

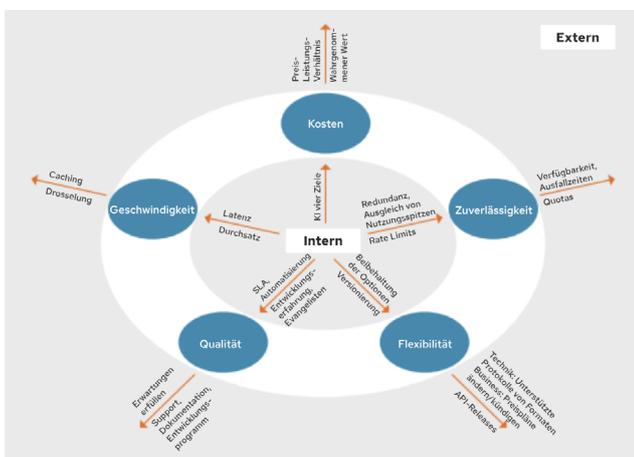


Abbildung 3. API Operations Donut

1. **Zuverlässigkeit.** Sicherstellung der API-Verfügbarkeit durch Redundanz oder Quotas und Ratenlimits. Diese sind auf die Geschäftsmodelle abgestimmt und verhindern Ausfallzeiten.
2. **Flexibilität.** Bieten Sie technische und geschäftliche Einführungsmöglichkeiten an, wie Kündigungen oder den Wechsel zwischen Preisoptionen. Denken Sie daran, dass eine größere Flexibilität zu Mehrkosten und einem höheren internen Aufwand führen kann.
3. **Qualität.** Konsistente Einhaltung der Entwicklererwartungen durch Service Level Agreements (SLAs) und optimierte Prozesse.
4. **Geschwindigkeit.** Geringe Latenzzeiten und hoher Durchsatz durch Techniken wie Drosselung und Caching, die möglicherweise auf Geschäftsmodelle abgestimmt sind.
5. **Kosten.** Optimieren Sie den Wert für Entwickler und minimieren Sie gleichzeitig interne Kosten, ohne die Qualität zu beeinträchtigen.

Bedenken Sie, dass mehrere Anbieter die technische Infrastruktur für viele dieser Herausforderungen im Bereich Operations bereitstellen, darunter auch Red Hat. Einen Anbieter zu nutzen ist oft eine kostengünstige Möglichkeit zur Lösung dieser Probleme, aber Ihre Strategie sollte gut durchdacht sein.

Wesentliche Komponenten der API-Verwaltung

Um ein API-Ökosystem effektiv zu verwalten, benötigen Sie ein besonderes Set von Komponenten. Obwohl diese Komponenten je nach API-Strategie variieren, bleiben die 3 Kernkomponenten gleich:

1. **Zugriffskontrolle.** Implementieren Sie Authentifizierungs- und Autorisierungssysteme, um Zugriff zu ermöglichen und eingehenden Datenverkehr zu identifizieren.
2. **Ratenlimits und Nutzungsrichtlinien.** Erzwingen Sie Quotas und Beschränkungen für den Datenverkehr, um die Auslastung vorhersagen zu können.
3. **Analytik.** Erfassen und analysieren Sie Verkehrsmuster zur Überwachung der API-Nutzung.

Überlegungen zu API-Operationen

Indem Sie sicherstellen, dass Ihre Strategie für den API-Betrieb mit den Gesamtstrategien Ihres Unternehmens harmoniert, können Sie Ressourcen entsprechend der Bedeutung der API zuweisen.

Berücksichtigen Sie bei Ihren Überlegungen zum Plan für API-Operationen folgende Bereiche:

1. **Zugriffskontrolle.** Definieren Sie, wer zugreift, Aktionen durchführt und Grenzen durchsetzt, um eine sicherheitsorientierte API-Nutzung zu gewährleisten.
2. **Metriken und Warnmeldungen.** Verschaffen Sie sich Transparenz mit Analysen, verwenden Sie maßgeschneiderte Metriken und stellen Sie Warnungen für die API-Performance ein.
3. **Umgang mit Nachfragespitzen.** Planen Sie die Infrastruktur mithilfe von Zugriffskontrollen sowie Richtlinien und richten Sie Ausweichmechanismen wie Drosselung oder den Ausgleich von Nachfragespitzen ein.
4. **Zuständigkeit für die API-Verfügbarkeit.** Eine klare Zuständigkeit für die API-Verfügbarkeit ist entscheidend, da diese direkt mit der Wertschöpfung und -erfassung verbunden ist.
5. **Umgang mit unerwünschter Nutzung.** Unterscheiden Sie zwischen erwarteter und unerwarteter unerwünschter Nutzung und steuern Sie diese durch proaktive Maßnahmen und Geschäftsbedingungen.
6. **Kommunikation.** Erstellen Sie einen klaren Kommunikationsplan für interne und externe Entwickler, um sie über geplante Ausfallzeiten, Wartungsarbeiten und API-Änderungen zu informieren.

Best Practice Nr. 5: Das Entwicklererlebnis ist der Schlüssel

Auch wenn Entwicklererlebnis so klingt, als ginge es dabei um das API-Design, geht es weit darüber hinaus. Betrachten Sie das Entwicklererlebnis als die Verpackung und Bereitstellung der API und nicht als die API selbst. Sie können eine wunderbar gestaltete containerisierte oder REST-API haben, aber wenn es schwierig ist, sich für den Zugriff anzumelden, Dokumente zu lesen und zu testen, haben Sie ein schlechtes Entwicklererlebnis geschaffen, das den Erfolg Ihrer API erheblich beeinträchtigen kann.

API-Erfolg ist mehr als tolles Design

Eine einfach und flexibel gestaltete API ist nutzlos, wenn sich die Entwickler nicht mit der API beschäftigen und sie schließlich einführen. Gleichzeitig hat ein gut durchdachtes API-Design erhebliche Auswirkungen auf das Entwicklererlebnis und die Einführung. Die Einführung ist ein wesentlicher Aspekt des Entwicklererlebnisses.

John Musser, einer der Pioniere auf dem Gebiet der APIs und API-Verwaltung, beschrieb mehrere Möglichkeiten zur Verbesserung der Einbindung, die immer noch aktuell sind:

1. Definieren Sie klar den Zweck der API.
2. Ermöglichen Sie eine einfache Anmeldung und kostenlosen Zugang.
3. Gestalten Sie die Preise transparent.
4. Bieten Sie eine sorgfältige Dokumentation.

Eine wichtige Kennzahl zur Verbesserung des API-Designs zwecks einfacher Nutzung ist die TTFHW-Metrik („Time to first hello world“), welche die anfängliche API-Interaktion erfasst. Ein breiterer Kontext wird durch die Zeitspanne bis zur ersten rentablen Anwendung (TTFPA) erfasst, wobei der Schwerpunkt auf Entwicklerprogrammen liegt. Diese Metrik ist etwas komplexer, denn die Interpretation, wann etwas rentabel ist, variiert je nach API und Geschäftsstrategie. Trotzdem ist TTFPA nützlich, weil Sie dabei über API-Operationen als Bestandteil des API-Programms nachdenken müssen.

Das Entwicklererlebnis umfasst 2 Grundsätze: hochwertiges Produkt- oder Dienstleistungsdesign und leichte Zugänglichkeit. Ein solides Programm zur Einbindung von Entwicklern - einschließlich Portal, Community, Evangelisten, Events und Messungen - verbessert das Engagement. Typische Metriken für das Entwicklerportal sind beispielsweise Seitenaufrufe, Anmeldungen, API-Traffic oder Supportanfragen. Events können an der Anzahl der Teilnehmenden, der API-Einführung bei Hackathons oder an Leads gemessen werden. Es ist hilfreich, Korrelationen zu erstellen, wie etwa „Hat ein Vortrag auf einer Veranstaltung zu mehr API-Anmeldungen geführt?“

Ein Entwicklerprogramm sollte die folgenden Elemente enthalten:

- **Aufbau einer Community.** Veranstaltungen wie Hackathons fördern Interaktionen und damit die Einführung.
- **Entwickler-Evangelisten.** Sie sind entscheidend für den Erfolg, da sie die Vorteile der API vermitteln.
- **Pilotpartner und Case Studies.** Early Adopters liefern Feedback und Case Studies.
- **Netzwerkpartner.** Partnerschaften verstärken die Einführung synergetisch.
- **Messungen.** Mit den API-Zielen übereinstimmende Metriken unterstützen ein effektives Management.
- **Kommunikation.** Portale, Newsletter, private Slack- oder Discord-Server.

Ein auf die Zielgruppe zugeschnittenes Entwicklerprogramm fördert das Engagement und das Erlebnis.

Überlegungen zur Evaluation des Entwicklererlebnisses

Ein gut durchdachtes Entwicklererlebnis kann diese dazu inspirieren, ihr Potenzial voll auszuschöpfen. Im Folgenden finden Sie 6 Überlegungen, die für die Bewertung des Entwicklererlebnisses unerlässlich sind und helfen, APIs zu entwickeln, die nicht nur Codezeilen, sondern Wege zu Kreativität und Produktivität darstellen.

1. **Erklärung des Wertes.** Verfassen Sie eine Kurzpräsentation, die Entwicklern den Wert der API vermittelt.
2. **TTFHW und TTFPA.** Bewerten und minimieren Sie die Zeitspanne bis zum ersten „Hallo Welt“ und die Spanne bis zur ersten rentablen Anwendung unter Berücksichtigung der Elemente des Entwicklererlebnisses (wie etwa Portale).
3. **Onboarding-Prozess.** Richten Sie das Onboarding an den API-Use Cases aus und stellen Sie einen frühen Erfolg der Entwickler sicher, indem Sie die Einfachheit beibehalten.
4. **Bereitstellung von Werten.** Stellen Sie sicher, dass die API genügend Wert bietet, um Entwickler anzuziehen und zu binden.
5. **Entwickler-Support.** Priorisieren Sie den Self-Service-Support durch Dokumentation, FAQ und Foren sowie zusätzlichen Mechanismen für komplexere Problemfälle.
6. **Unkonventionelle Nutzung.** Stellen Sie Support und Dokumentation für Entwickler mit nicht standardisierten Use Cases sicher.

Best Practice Nr. 6: Mehr als nur das 1x1 des Marketings

Es kann schwierig sein, APIs an Entwickler zu vermarkten, insbesondere wenn der angebotene Wert nicht mit den geschäftlichen oder technischen Anforderungen des Entwicklers übereinstimmt. APIs sollten wie andere Produkte vermarktet werden, im Einklang mit Segmentierung, Ausrichtung und Positionierung (STP). Effektives Marketing bringt die richtige API mit den richtigen Entwicklern zusammen, die sich am API-Wert und STP-Prinzipien orientieren.

1. **Segmentierung.** Kategorisieren Sie Kunden als interne Nutzende, Partner, Endnutzende oder externe Entwickler. Unterteilen Sie den riesigen Entwicklermarkt nach der Jobs-to-be-done-Methode, um ein effektives Engagement zu erreichen.
2. **Ausrichtung.** Bewerten Sie die Attraktivität des Segments anhand von Zugänglichkeit, Substanz und Differenzierung. Wählen Sie vielversprechende Segmente aus und passen Sie Ihre Marketingtaktiken entsprechend an.
3. **Positionierung.** Heben Sie Ihre API hervor, indem Sie auf spezifische Bedürfnisse eingehen, Probleme lösen und Vorteile für ausgewählte Entwicklersegmente bieten.

Für den Erfolg ist es entscheidend, dass das Entwicklererlebnis beim Marketing im Vordergrund steht. Strategien wie Entwickler-Evangelisten, robuste Entwicklerportale, Hackathons und andere Veranstaltungen schaffen solide Beziehungen zu Entwicklern.

Überlegungen zum API-Marketing

1. **Zielgruppen.** Setzen Sie Prioritäten bei den wichtigsten Nutzergruppen und entwickeln Sie Ihre API schrittweise basierend auf internen Nutzenden, Partnern, Kunden oder der Öffentlichkeit weiter.
2. **Auswahl von Fachleuten.** Wählen Sie Experten und Expertinnen aus, die dem Leistungsversprechen Ihrer API gerecht werden, und berücksichtigen Sie dabei Bereiche wie Technik, Support, Vertrieb und Produktmanagement.
3. **Event-Strategie.** Wählen Sie Event-Typen (horizontal oder vertikal, global oder lokal, Konferenz oder Hackathon) aus, die zu Ihren API-Zielen passen.
4. **Hackathon-Eignung.** Bewerten Sie die Relevanz eines Hackathons (Anmeldungen, SDK-Downloads, Bewerbungen, Rekrutierung, Branding) und planen Sie entsprechend.
5. **Internes Marketing.** Sichern Sie sich Unterstützung in verschiedenen Bereichen, stimmen Sie sich mit der Marketingabteilung ab und kommunizieren Sie die Vorteile gegenüber Produktteams und Kunden.

Best Practice Nr. 7: Denken Sie an Abschaltung und Änderungsmanagement von APIs

Ratschläge zum Thema API konzentrieren sich oft stark auf das Entwickeln, Erstellen und Betreiben von APIs. Eine der kritischsten Aufgaben, die oft übersehen wird, findet allerdings erst viele Monate nach der Einführung und Betriebsaufnahme statt: die Verwaltung von Aktualisierungen sowie das Stilllegen von APIs.

Unterbrechungen durch abrupte Veränderungen können das Vertrauen untergraben und erhebliche Kosten verursachen, insbesondere bei unbekanntem Entwicklern, bei der Genehmigung mobiler Anwendungen oder bei Geräten, die nicht aktualisiert werden können.

API-Änderungen werden in der Regel als „Non-Breaking Change“ oder „Breaking Change“ bezeichnet. „Non-Breaking Changes“ beinhalten neue Methoden und Verbesserungen, während „Breaking Changes“ das Entfernen, Modifizieren oder das vollständige Stilllegen bezeichnen. Hauptversionsnummern und Migrationspläne adressieren Breaking Changes, Nebenversionen kümmern sich um Non-Breaking Changes. Es kann schwierig sein, sicherzustellen, dass Änderungen nicht zum Ausfall bestimmter Anwendungen führen. Daher empfehlen wir dringend, dass alle Änderungen an der API, einschließlich der als nicht schädlich eingestuft, folgendermaßen durchgeführt werden:

- Durch Bereitstellung eines Test-Endpunkts mit der neuen Version vor dem Launch
- Durch Versand einer E-Mail oder einer anderen Mitteilung an die Entwickler, in der über die Änderung informiert wird und die den Zeitplan sowie Einzelheiten enthält

Eine effiziente Kommunikation und transparente Verträge sind unerlässlich. Teilen Sie Problemdetails mit, halten Sie Zusagen ein und geben Sie die Dauer des Versionssupports an. So erfordert beispielsweise die Stilllegung einer API einen strukturierten Ansatz, einschließlich einer ausreichenden Vorankündigung, Medienberichterstattung, Migrationsplänen und bei Bedarf Tools für den Datenexport.

Ein Migrationsplan erleichtert API-Updates:

1. Stellen Sie eine neue Version zum Testen bereit
2. Benachrichtigen Sie die Nutzenden über die Einstellung der alten Version
3. Unterstützen Sie die Nutzenden bei der Umstellung
4. Nehmen Sie die alte Version außer Betrieb

Zu einer umfassenden API-Verwaltung gehört es, Aktualisierungen und Stilllegungen zu antizipieren, effektiv zu kommunizieren und Vertrauen durch transparente Maßnahmen zu erhalten.

Überlegungen zur Lebensdauer von APIs

- 1. Sicherstellung der Einhaltung Ihrer Kundengarantie.** Der wohl wichtigste Punkt Ihres API-Programms: Welchen Grad an Servicestabilität sind Sie bereit, Ihren Nutzenden zu garantieren? Definieren Sie die Verpflichtung zur Servicestabilität für die Nutzenden. Diese Verpflichtung beeinflusst die Einführung.
- 2. Change- und Breaking Change-Prozess.** Festlegung von Releaseverfahren zur Aufrechterhaltung der Garantie. Bestimmen Sie die beteiligten Parteien und die Schritte zur Genehmigung.
- 3. Kommunikation von Änderungen.** Erkennen, dokumentieren und übermitteln Sie Änderungen mithilfe von API-Definitionsformaten. Sorgen Sie für Kompatibilität und klare Kommunikation.
- 4. Versionsverwaltung.** Überwachen Sie die Nutzung älterer Versionen anhand von Entwickler- und Nutzenden-IDs. Etablieren Sie ein Verfahren für die Stilllegung, um Problemen vorzubeugen.
- 5. Produktausrichtung.** Koordinierung der API-Entwicklung mit den entsprechenden Produktänderungen unter Berücksichtigung der Kundenverpflichtungen und erforderlicher Anpassungen.

Nachhaltigkeit Ihrer API-Strategie

Diese Best Practices helfen Ihnen bei der Definition, Umsetzung und Verbesserung Ihrer API-Strategie. Sie sollen Sie dabei unterstützen, Ihre Strategie anzupassen und neue API-Möglichkeiten zu entdecken. Es ist gut möglich, dass viele der Fragen in den vorangegangenen Abschnitten sehr detailliert beantwortet werden müssen und dass sich im Laufe der Zeit neue Chancen oder Risiken ergeben. Beispiel:

- Gibt es Möglichkeiten, den Wert der API für den Kunden zu erhöhen?
- Haben wir ausreichend für die Zukunft vorgesorgt?
- Ist das Leistungsversprechen attraktiv genug, um langfristig Entwickler anzuziehen?

So wie sich die IT-Landschaft weiterentwickelt, müssen auch Ihre API und Angebote sich anpassen. Zu den 4 wichtigsten Faktoren für Veränderungen gehören:

1. **Branchenkräfte.** Neue Mitbewerber oder Dienste könnten Ihre API ersetzen.
2. **Marktkräfte.** Sich wandelnde Nutzendenanforderungen oder Bedingungen im Marktsegment.
3. **Makroökonomische Kräfte.** Globale Marktveränderungen beeinflussen die Budgets der Nutzenden.
4. **Trends.** Sich entwickelnde Technologien oder gesetzliche Vorgaben.

Erstellen eines langlebigen API-Programms

Ein API-Programm, das solide Use Cases mit potenziellen Kunden verbindet, kann erfolgreich beginnen, wachsen und sich weiterentwickeln. Diese erfolgreichen API-Programme schaffen Raum für Innovationen durch geplante API-Flexibilität und indem sie potenziell negative Nutzungsszenarien mithilfe von Design und Abläufen lösen. Sie müssen außerdem sicherstellen, dass Ihre API-Geschäftsbedingungen Ihnen die notwendigen Instrumente zum Vorgehen gegen unerwartetes Verhalten geben.

In den folgenden Fällen müssen Sie Ihre API-Strategie wahrscheinlich umstrukturieren:

- Der Wert wird durch die Abhängigkeit von unerwarteten Innovationen überschattet. Eine robuste Infrastruktur ist für den Erfolg dieses Ansatzes entscheidend.
- Das Fehlen überzeugender Use Cases für Sie und die Nutzenden deutet auf die Notwendigkeit eines anderen Ansatzes hin.
- Interne Zweifel bezüglich negativen Verhaltens bleiben bestehen, was auf unzureichende Maßnahmen oder Kommunikation hinweist.
- Häufige Kompromisse oder Zweckentfremdung der API offenbaren eine Diskrepanz zwischen beabsichtigtem und tatsächlichem Wert.

APIs existieren nicht in einem Vakuum

Die sich ständig weiterentwickelnde Technologielandschaft hat eine Reihe leistungsfähiger Tools zur Entwicklung cloudnativer Anwendungen hervorgebracht, darunter Kubernetes, Red Hat® OpenShift® und Red Hat OpenShift Service Mesh, die durch ihr Zusammenspiel die Konnektivität verbessern und die Entwicklung außergewöhnlicher Anwendungen beschleunigen. Inmitten dieser Fortschritte bleiben jedoch die Grundprinzipien einer robusten API-Strategie und die Bedeutung eines effektiven API-Verwaltungssystems unverändert. Suchen Sie nach einer API-Verwaltungslösung, die nicht nur Ihrer übergeordneten Strategie entspricht, sondern auch ein ausgeprägtes Verständnis für diese Plattformen oder eine konsistente Integration mit ihnen aufweist, um die Umsetzung Ihrer strategischen Vision zu erleichtern.

Warum sollten Sie Red Hat für die Entwicklung und Verwaltung Ihrer APIs wählen?

Red Hat steht für einen strategischen Ansatz, bei dem das API-Design im Vordergrund steht, um den Erfolg Ihres API-Programms sicherzustellen. Dieser Ansatz umfasst den gesamten API-Lifecycle, von der anfänglichen Planung und dem Design, darunter das Mapping von Ressourcen und die Modellierung von Geschäftsszenarien, bis hin zur Verwaltung der APIs, die die Zugriffskontrolle, den Zugriff auf die APIs und die Analyse der Nutzung einschließt.

Red Hat bietet die speziell für Hybrid Cloud-Umgebungen entwickelte API-Verwaltungslösung [Red Hat 3scale API Management](#) an, um Entwicklungsteams bei diesen Best Practices des auf API-Design fokussierten Ansatzes zu unterstützen, über den Sie in unserem API-Benutzerhandbuch gelesen haben. Unsere verteilte, schlanke, containerbasierte und cloudnative Lösung ermöglicht den Umgang mit großen Workloads und fördert einen sicherheitsorientierten Ansatz sowie die Zusammenarbeit. Die gemeinsame Nutzung und Kontrolle des Zugriffs auf Dienste, Ressourcen, Anwendungen und Unternehmenssysteme in Public- und Private Cloud-Umgebungen stärkt Ihr Unternehmen in komplexen Netzwerken.

APIs existieren jedoch nicht im Vakuum, und für die Umsetzung einiger dieser Best Practices benötigen Sie eine vollständige Anwendungsplattform für die Entwicklung cloudnativer Anwendungen. Daher ist Red Hat 3scale API Management Teil des [Red Hat Application Foundations](#) Portfolios. Dabei handelt es sich um eine Sammlung von Middleware-Tools, die Funktionen wie Integration, API-Design, API-Governance und -Registrierung, Streaming und Messaging umfassen und für die Zusammenarbeit mit [Red Hat OpenShift Container Platform](#) konzipiert sind. Durch den gemeinsamen Einsatz von Red Hat OpenShift und Red Hat Application Foundations verfügen Unternehmen über eine komplette cloudnative Anwendungsplattform, mit der sie Nutzenden neue Software effizienter und sicherer zur Verfügung stellen und gleichzeitig strategische Fähigkeiten und Vorteile für ihr Unternehmen erschließen können.

Verbessern Sie Ihr Konzept für die Entwicklung cloudnativer Anwendungen mit Red Hat 3scale API Management, Red Hat Application Foundations und Red Hat OpenShift und erleben Sie die Zukunft der optimierten, effizienten und kooperativen Anwendungsentwicklung.

Fazit

Wir hoffen, dass diese Best Practices Ihnen bei Ihren Entscheidungen oder Fragestellungen helfen. Die Entwicklung Ihrer API-Strategie beginnt mit einem klaren, konsistenten und unternehmensweiten Verständnis des Werts Ihrer API. Dieser Wert und die Geschäftsziele Ihrer API sind jedoch nicht statisch. Umstände können sich ändern und Ihre API-Strategie muss sich mit ihnen verändern.

Basierend auf unseren Erfahrungen bei der Zusammenarbeit mit unseren Kunden und Beobachtungen in der API-Economy, können wir einige Merkmale eines hochwertigen API-Programms zusammenfassen:

- Die Nutzenden führen eine API ein, weil sie für sie von Wert ist. Die API erfüllt eine wichtige Aufgabe, indem sie Schwierigkeiten beseitigt oder Gewinne erzielt.
- Eine API bietet den Nutzenden einen kontinuierlichen Mehrwert, indem sie ein Leistungsversprechen und eine Strategie bietet, die sich mit der Umgebung ändert.
- Eine API ist unternehmensintern wertvoll. Sie leistet einen wichtigen Beitrag beim Erreichen bedeutender Unternehmensziele.
- Möglichst viele und im Idealfall alle Stakeholder sind zufrieden.

Erfahren Sie mehr zu [Red Hat 3scale for API Management](#) und [Red Hat API Management Ressourcen](#).