

Mit Empfehlung von **IBM**

Limitierte Auflage von IBM

APIs

FÜR DUMMIES®

Sie erfahren:

- Wie Sie die Leistungsfähigkeit von APIs für Business-Herausforderungen nutzen können
- Warum moderne APIs ein Produkt darstellen
- Wie Sie entscheiden, welche APIs Sie bereitstellen oder nutzen möchten
- Wie Sie eine effektive API-Technologieplattform aufbauen



Claus T. Jensen

APIs
FÜR
DUMMIES®

Limitierte Auflage von IBM

von Claus T. Jensen

WILEY

APIs For Dummies®, limitierte Auflage von IBM

Veröffentlicht von
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

Copyright © 2016 John Wiley & Sons, Inc.

Diese Veröffentlichung darf ohne vorherige schriftliche Genehmigung des Herausgebers weder vervielfältigt, in einem Abfragesystem abgelegt, noch in irgendeiner Form oder auf irgendeine Weise übertragen werden (elektronisch, maschinell, durch Kopieren, Aufzeichnen, Scannen oder auf andere Weise). Ausgenommen hiervon sind die Informationen in den Abschnitten 107 und 108 des United States Copyright Act von 1976. Genehmigungsanfragen an den Herausgeber sind an folgende Adresse zu richten: Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030; Tel. (201) 748-6011, Fax (201) 748-6008; oder online unter <http://www.wiley.com/go/permissions>.

Marken: Wiley, das Wiley-Logo, For Dummies, das Dummies Man-Logo, The Dummies Way, Dummies.com, Making Everything Easier und damit zusammenhängende Handelsformen sind Marken oder eingetragene Marken von John Wiley & Sons, Inc. und/oder den angeschlossenen Unternehmen in den USA und anderen Ländern. Sie dürfen ohne schriftliche Genehmigung nicht verwendet werden. IBM und das IBM Logo sind eingetragene Marken der International Business Machines Corporation. Alle anderen Marken sind Eigentum der jeweiligen Markeninhaber. John Wiley & Sons, Inc., steht nicht im Zusammenhang mit einem in dieser Veröffentlichung genannten Produkt oder Anbieter.

HAFTUNGSBESCHRÄNKUNG/HAFTUNGSAUSSCHLUSS: HERAUSGEBER UND AUTOR GEBEN KEINE ZUSICHERUNGEN ODER GEWÄHRLEISTUNGEN IM HINBLICK AUF DIE RICHTIGKEIT ODER VOLLSTÄNDIGKEIT DER INHALTE IN DIESEM DOKUMENT. AUSSERDEM WIRD KEINE HAFTUNG IM HINBLICK AUF DIE EIGNUNG FÜR EINEN BESTIMMTEN ZWECK ÜBERNOMMEN. AUS VERKAUFS- ODER PROMOTION-MATERIALIEN KANN KEINE GEWÄHRLEISTUNG ABGELEITET ODER AUSGEWEITET WERDEN. DIE EMPFEHLUNGEN UND STRATEGIEN IN DIESEM DOKUMENT EIGNEN SICH MÖGLICHERWEISE NICHT FÜR JEDE SITUATION. DER VERKAUF DES DOKUMENTS ERFOLGT UNTER DER ANNAHME, DASS DER HERAUSGEBER KEINE RECHTLICHEN, BUCHHALTERISCHEN ODER ANDEREN PROFESSIONAL SERVICES LEISTEN MUSS. FALLS UNTERSTÜTZUNG DURCH EXPERTEN BENÖTIGT WIRD, SOLLTEN DIE SERVICES EINER KOMPETENTEN PERSON IN ANSPRUCH GENOMMEN WERDEN. FÜR SCHÄDEN, DIE SICH Daraus ERGEBEN, SIND WEDER DER HERAUSGEBER NOCH DER AUTOR VERANTWORTLICH. DIE TATSACHE, DASS IN DIESEM DOKUMENT AUF EIN UNTERNEHMEN ODER EINE WEBSITE ALS ZITAT UND/ODER MÖGLICHE INFORMATIONSQUELLE VERWIESEN WIRD, BEDEUTET NICHT, DASS DER AUTOR ODER DER HERAUSGEBER DIE INFORMATIONEN ODER EMPFEHLUNGEN DES JEWEILIGEN UNTERNEHMENS ODER AUF DER JEWEILIGEN WEBSITE BEFÜRWORTET. DER LESER SOLLTE ZUDEN BERÜCKSICHTIGEN, DASS SICH DIE IN DIESEM DOKUMENT AUFGEFÜHRTEN ADRESSEN VON WEBSITES IM ZEITRAUM ZWISCHEN DER ERSTELLUNG UND DER VERÖFFENTLICHUNG DES DOKUMENTS GEÄNDERT HABEN KÖNNEN ODER NICHT MEHR VORHANDEN SIND.

Wenn Sie allgemeine Informationen über unsere anderen Produkte und Services wünschen oder erfahren möchten, wie ein individuelles *For Dummies*-Dokument für Ihr Unternehmen erstellt wird, wenden Sie sich an das Business Development Department in den USA unter 877-409-4177. Sie können auch an info@dummies.biz schreiben oder folgende Website besuchen: www.wiley.com/go/custompub. Weitere Informationen zur Lizenzierung der *For Dummies*-Marke für Produkte oder Services erhalten Sie über BrandedRights&Licenses@Wiley.com.

ISBN: 978-1-119-19339-5 (pbk); ISBN: 978-1-119-19354-8 (ebk)

Hergestellt in den USA

10 9 8 7 6 5 4 3 2 1

Danksagung des Verlags

An der Veröffentlichung dieses Buches waren unter anderem die folgenden Personen beteiligt:

Projektedaktion: Carrie A. Johnson

Entwicklungslektorat: Kathy Simpson

Redaktionelle Leitung: Rev Mengle

Geschäftsfeldentwicklung: Sue Blessing

Produktionskoordination: Melissa Cossell

Inhalt



Einleitung.....	1
Über dieses Buch	1
Symbole in diesem Buch	2
Was nicht in diesem Buch steht.....	3
Kapitel 1: Die Anatomie einer API	5
Was eine API nicht ist	5
APIs als Produkt	5
Von APIs zur API Economy	6
Was Entwickler wollen.....	7
Kapitel 2: APIs verwalten – oder auch nicht	9
Was es heißt, eine API zu verwalten.....	9
API-Verantwortliche im Business.....	10
IT-Betrieb	10
API-Designer	11
Warum API-Governance notwendig ist	12
Entscheidungen von API-Anbietern.....	12
Entscheidungen von API-Nutzern	13
Ein Plädoyer für nicht verwaltete APIs	13
Kapitel 3: Die Beschaffenheit guter APIs	15
APIs im Vergleich mit Rennwagen	15
Ein Plädoyer für angepasste APIs.....	16
Ein paar Überlegungen zu APIs und Services	17
APIs im Vergleich zu Services	17
APIs und Services im Team.....	19
Gutes API-Design erkennen.....	19
Kapitel 4: API-Einstiegspunkte	21
Daten monetarisieren	22
Innovationsfreiheit	23
Mobil in zehn Minuten.....	26
Leben in einer hybriden Welt	29
Programmieren Sie Ihre Welt.....	32

Kapitel 5: API- und Integrationsmiddleware 35

API-Middleware ist mehr als „nur ein weiterer ESB“	36
Die (reproduzierbare) Integrationstopologie	37
API und Service Economy: Referenzmodell	40

**Kapitel 6: Zehn Dinge, die Sie über APIs
wissen sollten 41**

Omnichannel-Erfahrungen fördern die Nachfrage nach APIs	41
APIs sind Business-Produkte	42
Business-Design als Herausforderung, die alle betrifft	42
Erkenntnisse durch API-Instrumentierung	43
Nicht alle APIs sind REST-APIs	43
Jede API braucht einen Verantwortlichen im Business	43
APIs müssen versioniert werden	44
APIs lassen sich mit Richtlinien leicht kontrollieren	44
APIs haben eine Schattenseite	44

Einleitung



APIs sind ein brandaktuelles Thema, das von Unternehmen, IT-Verantwortlichen wie Entwicklern gleichermaßen engagiert diskutiert wird. Ein Großteil des öffentlichen Hypes dreht sich dabei um offene, öffentliche APIs. Heutzutage nicht über eine öffentliche API zu verfügen, ist in gewisser Weise so, wie in den späten 1990er-Jahren keine Webseite zu haben. Für viele Unternehmen sind öffentliche APIs tatsächlich jedoch das kleinste ihrer geschäftlichen Anliegen. Zu den wichtigeren Fragestellungen gehören der Aufbau von Omnichannel-Lösungen, die Fähigkeit, Innovationen schneller zu realisieren als die Konkurrenz, der Wandel hin zu einem mobilen Unternehmen oder die Arbeit in einer hybriden Cloudumgebung.

APIs sind elementare Hilfsmittel für diese und weitere Initiativen, was der Grund dafür ist, dass sich so viele unterschiedliche Akteure für sie interessieren. Aber was ist eine API tatsächlich? Inwiefern unterscheidet sie sich von einer Anwendungsprogrammierschnittstelle alten Typs? Und warum sollte Sie das überhaupt interessieren? Im Prinzip steht das Akronym API für *Application Programming Interface, also Anwendungsprogrammierschnittstelle*, aber die Vorstellung davon, was API bedeutet, hat sich signifikant weiterentwickelt. Die APIs von heute unterscheiden sich erheblich von den Anwendungsprogrammierschnittstellen früherer Tage. In diesem Buch erfahren Sie, was diese Veränderung mit sich bringt.

Über dieses Buch

Moderne Geschäftsfelder müssen in puncto Innovation und Integration umdenken. Dieses Buch soll Ihnen helfen, die Leistungsfähigkeit von APIs zu nutzen, um aktuelle geschäftliche Herausforderungen zu meistern, angefangen bei sich ändernden Geschäftsmodellen bis hin zum Ergreifen der Chancen, die eine Welt aus Geräten und Sensoren bietet. Die Nutzung der Leistungsfähigkeit von APIs beinhaltet sehr viel mehr als die schlichte Monetarisierung von Daten. Egal, ob Sie API-Anbieter oder API-Nutzer sind, Sie müssen intelligente Business- und IT-Entscheidungen treffen. In diesem Buch wird zuerst der Grundcharakter moderner APIs definiert. Anschließend führt es Sie durch verschiedene notwendige Entscheidungen, angefangen bei der Frage, welche APIs Sie anbieten

oder nutzen möchten, bis hin zur Frage, wie Sie eine erfolgreiche API-Technologieplattform aufbauen können.

Darüber hinaus wird an verschiedenen Stellen dieses Buchs auf eine Reihe wichtiger Leitsätze eingegangen. Hierzu zählen:

- ✓ **Betrachten Sie eine API als ein Produkt.** Eine API ist etwas, was Sie mit einer bestimmten Zielgruppe unter bestimmten Bedingungen teilen möchten.
- ✓ **Halten Sie sich an das Prinzip „Früh testen, schnell lernen und ohne großen Aufwand erweitern“.** Eine experimentierende Haltung zu APIs ist in den meisten Anwendungsfällen effektiv und viele APIs sind von Natur aus angepasst.
- ✓ **Nutzen Sie APIs stets als Grenze Ihrer Domäne.** Hierdurch behalten Sie die Kontrolle und schaffen Transparenz in Bezug auf den ein- und ausgehenden Datenverkehr.
- ✓ **API-Plattformen sind aufgabenorientiert.** Erstellung, Nutzung und Teilen von APIs sollte kinderleicht und absolut sicher sein.
- ✓ **Die Monetarisierung öffentlicher APIs ist nicht der einzige Einstiegspunkt.** Viele Unternehmen nutzen APIs für die Zusammenarbeit und Innovationen in ihren IT- und Business-Ökosystemen.

Symbole in diesem Buch

Wie alle *For Dummies*-Bücher werden auch in diesem Buch am Seitenrand einige Symbole verwendet. Diese Symbole haben folgende Bedeutung:



Das Symbol „Tipp“ weist auf nützliche Informationen hin.



Alles, was mit dem Symbol „Nicht vergessen“ gekennzeichnet ist, sollten Sie sich merken.



Wenn Sie sich nicht für die technischen Details interessieren, können Sie die Abschnitte mit dem Symbol „Achtung Technik“ getrost überspringen.



Bei dem Symbol „Warnung“ müssen Sie aufpassen. Es weist auf Dinge hin, die schädlich für Ihr Unternehmen sein können.

Was nicht in diesem Buch steht

Dieses Büchlein kann nicht jedes Detail eines Themas abdecken. Wenn Sie an Informationen interessiert sind, die über die in diesem Buch bereitgestellten Informationen hinausgehen, sollten Sie die folgenden Links nutzen:

- ✓ developer.ibm.com/api/blog: IBM Blog zum Thema „APIs“
- ✓ www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg248188.html?Open: Ausführliche Informationsquelle zu Integrationsmiddleware
- ✓ ibm.com/soa: Hier können Sie das E-Book *SOA-Designprinzipien for Dummies* herunterladen

Kapitel 1

Die Anatomie einer API

In diesem Kapitel

- ▶ Was eine API nicht ist
- ▶ Wie Entwickler und Nutzer APIs verwenden möchten
- ▶ APIs kategorisieren
- ▶ APIs als Teil des Geschäftsmodells etablieren

Moderne APIs sind eine flexible Methode, um eigene Funktionalität für eine Zielgruppe außerhalb des eigenen Teams zur Verfügung zu stellen. Richtig gemacht bieten APIs einem Unternehmen die Möglichkeit, Innovationen zu beschleunigen und neue Zielgruppen zu erreichen. Das ist der Mehrwert von APIs. Aber was ist ihr Grundcharakter und welche Fragen sollten Sie sich stellen, wenn Sie sich mit APIs befassen möchten? In diesem Kapitel wird versucht, diese Fragen zu beantworten.

Was eine API nicht ist

Manchmal lässt sich etwas am einfachsten dadurch erklären, dass man beschreibt, was es nicht ist. In diesem Sinne hier nun eine kurze Liste dessen, was APIs nicht sind:

- ✓ **Ein Stück Software** Software ist keine API (kann sich jedoch als API präsentieren, um die Nutzung ihrer Funktionalität einfacher zu machen).
- ✓ **Eine Benutzeroberfläche** Eine Benutzeroberfläche ist keine API (kann jedoch auf einer API aufsetzen).
- ✓ **Ein Server:** Ein Server ist keine API (kann jedoch eine oder mehrere APIs hosten, die die Daten und Funktionen zugänglich machen, die vom Server bereitgestellt werden).

APIs als Produkt

Betrachten Sie eine API als ein Produkt. Gestalten Sie sie so, dass sie für den zukünftigen Nutzer interessant ist – und sich dadurch verkaufen lässt. Es spielt keine Rolle, ob der Nutzer tatsächlich für die

API bezahlt oder ob er sich außerhalb des Unternehmens befindet oder einem anderen Team innerhalb des Unternehmens angehört. Ihr Ziel ist es, dass er Ihre API verwendet, da hierdurch für Sie und ihn ein Mehrwert entsteht.

Die produkthafte Beschaffenheit von APIs ist eine wesentliche Voraussetzung für ihre Leistungsfähigkeit. Dadurch unterscheiden sie sich außerdem erheblich von Anwendungsprogrammierschnittstellen alten Stils, die ein Stück Software darstellten, die erstellt und implementiert wurden. Eine API modernen Typs ist ein Paket aus Funktionalität, die für eine Zielgruppe interessant ist, und zwar unabhängig davon, ob eine bestimmte Softwareressource auf Ihrem Back-End ausgeführt wird. Somit sind moderne APIs, obwohl sie eine definierte Programmschnittstelle enthalten, bewusst aus der Sicht des zukünftigen Nutzers entwickelt.



Da eine API ein Produkt ist, sollten Sie sich vor der Entwicklung einer API die folgenden Schlüsselfragen stellen:

- ✓ Wer sind die zukünftigen Nutzer?
- ✓ Wie möchten Sie diese Nutzer erreichen?
- ✓ Unter welchen Bedingungen können Nutzer diese API verwenden?

Von APIs zur API Economy

Die API Economy beginnt, wenn APIs Teil des Geschäftsmodells werden. Öffentliche und Partner-APIs sind eine strategische Kraft für verschiedene Geschäftsmodelle, beispielsweise das Geschäftsmodell von Twitter und Amazon.

Die Twitter-APIs verzeichnen beispielsweise mühelos zehnmal so viel Datenverkehr wie die Twitter-Website selbst. Das Geschäftsmodell des Unternehmens konzentriert sich bewusst auf die Vermittlung von Tweets, sodass jeder, der möchte, die Endbenutzererfahrung dafür gestalten kann.

Von Beginn an gehörte es zum Geschäftsmodell von Amazon, nicht nur als Internethändler, sondern auch als universelles Handelsportal aufzutreten. Die Händlerplattform von Amazon setzt bewusst auf APIs auf, die den problemlosen Einstieg neuer Händler ermöglichen.

APIs als treibende Kraft von Unternehmensnetzwerken sind keine neue Erfindung. Banken verwenden bereits seit Jahrzehnten klar definierte APIs als Basis für Zahlungsinfrastrukturen und Abrechnungsstellen. Moderne APIs werden jedoch explizit für ein offenes Ökosystem (intern oder extern) und nicht für geschlossene, private

Netzwerke entwickelt. Darüber hinaus sind die Nutzungsmodelle für APIs schwerpunktmäßig auf einfache Nutzung anstatt auf einfache Erstellung ausgerichtet.

Mitunter wird der Begriff *Geschäfts-APIs* für alle APIs modernen Stils verwendet. Der Begriff passt insofern, als APIs, als Produkt betrachtet, ein wesentlicher Bestandteil Ihrer Geschäftsstrategie sein sollten. Machen Sie sich jedoch bewusst, dass die Einführung einer öffentlichen oder Partner-API nicht die einzige Möglichkeit ist, um APIs in Ihr Geschäftsmodell zu integrieren. Es gibt viele Anwendungsfälle für intern genutzte APIs; der gängigste dieser Anwendungsfälle ist vermutlich die Notwendigkeit zur Bereitstellung einer unterscheidbaren *Omnichannel-Benutzererfahrung*.



Ganz gleich, ob Ihr Unternehmen mit dem Internet groß geworden ist oder schon seit 100 Jahren existiert, Sie leben im Zeitalter von Cloud, Analysetechnologien, Mobilität und Social Computing, in dem Omnichannel das Gebot der Stunde ist. Um sich von der Konkurrenz abzuheben, müssen Sie Ihren Kunden ein Benutzererlebnis bieten, das sie unmittelbar fesselt und für Sie einnimmt. Um ein solches Benutzererlebnis schaffen zu können, benötigen Sie Platz für Experimente und Innovationen. Nutzen Sie die Chance und orientieren Sie sich am Prinzip „früh testen, schnell lernen und ohne großen Aufwand erweitern“.

Was Entwickler wollen

Entwickler möchten APIs verwenden, um zu experimentieren und Neues zu schaffen. Für sie geht es bei der Wiederverwendung um schnellerer Bereitstellung, bei der gemeinsamen Nutzung um Zweckmäßigkeit und bei der Kapselung darum, wenig Neues lernen zu müssen. Die Frage, wie APIs konstruiert wurden, interessiert sie deutlich weniger als die Frage, wie leicht die APIs zu nutzen sind.



Einfache Nutzung bezieht sich nicht allein darauf, wie die API aussieht. Für einen Entwickler, der sich mit APIs auskennt, bedeutet einfache Nutzung auch, dass das Auffinden und die zur Verwendung der API notwendige Registrierung einfach sein muss. Außerdem muss klar erkennbar sein, inwieweit einer API in unternehmenskritischen Lösungen vertraut werden kann.

Im Idealfall basiert ein API-Ökosystem stark auf einer Community. Eine leistungsfähige API-Community zeigt Entwicklern genau, welche APIs ihnen für ihre aktuellen Aufgaben zur Verfügung stehen. Außerdem bietet sie die Self-Service-Registrierung und Vorabgenehmigungen für die APIs, die der Community zugänglich sind. Die Social-Features der Community ermöglichen es den Personen, eine bestimmte API zu empfehlen oder abzulehnen, und mittels nutzerorientierter Analyse

lässt sich das erwartete Einsatzverhalten jeder API zeigen, für die sich ein Entwickler interessiert. Diese Möglichkeiten gehören traditionell nicht zur IT-Governance, sind jedoch zentrale Funktionen von intelligenten API-Verwaltungslösungen. Intelligente API-Verwaltungslösungen bieten API-Anbietern außerdem einen zusätzlichen Nutzen, da sie die Erstellung von APIs sowie die Steuerung ihres Laufzeitverhaltens vereinfachen (was in Kapitel 2 ausführlicher erläutert wird).

Vier Kategorien von APIs

Wenn Sie sich erstmalig mit dem Thema „APIs“ beschäftigen, kann die Entscheidung, welche APIs Sie erstellen möchten, eine respektieinflößende Aufgabe sein. Eine gute API sollte eine Veränderung für das Unternehmen bewirken.

In dieser Zwischenbemerkung wird eine Herangehensweise erläutert, die sich bei IBM in der Praxis als nützlich erwiesen hat. Als Einstieg hat sich folgende Frage bewährt: „Welche geschäftlichen Gegebenheiten möchte ich verbessern und wie kann ich das erreichen?“ Die Beantwortung dieser Frage führt höchstwahrscheinlich dazu, dass Sie Ihre ersten APIs aus einer der folgenden vier Kategorien auswählen:

✓ **Erkennungs-APIs:** Erkennungs-APIs helfen Ihnen, Chancen für die Gewinnung oder Einbindung von Kunden, Mitarbeitern, Partnern und Geräten zu identifizieren. Sie schließen Mechanismen wie mobile Standorterkennung, Sensorüberwachung, vorhersagende Analyse und Beobachtung menschlichen Verhaltens ein.

✓ **Anreicherungs-APIs:** Anreicherungs-APIs verbessern das Verständnis eines Sachverhalts durch historische Daten aus Customer-Relationship-Management-Systemen (CRM), Kontodatensätzen, demographischen Analysen, Patientenakten und ähnlichem mehr.

✓ **Wahrnehmungs-APIs:** Wahrnehmungs-APIs liefern dynamischen Kontext für die aktuelle Situation und ermöglichen es Ihnen zu verstehen, was die Menschen bewegt, die Sie ansprechen möchten. Beispiele hierfür sind Social-APIs (Menschen, die Zukunftspläne oder aktuelle Interessen teilen) und Sensoranalysen (zum Gesamtzustand eines Systems, beispielsweise der weltweite Ressourcenverbrauch oder das Verkehrsaufkommen).

✓ **Aktions-APIs:** Aktions-APIs ermöglichen es Ihnen, nahezu in Echtzeit Schritte einzuleiten. Beispiele für aktionsartige Schnittstellen sind Push-Benachrichtigungen, instrumentierte Geräte und Systeme für die Verwaltung menschlicher Aufgaben.

Kapitel 2

APIs verwalten – oder auch nicht

In diesem Kapitel

- ▶ Funktionen und Aufgaben der API-Verwaltung im Überblick
- ▶ Warum API-Governance notwendig ist
- ▶ Wann nicht verwaltete APIs verwendet werden sollten

Ein wesentlicher Bestandteil vieler Diskussionen zum Thema „APIs“ ist die Frage der API-Verwaltung. Obwohl APIs keine Softwareressourcen im herkömmlichen Sinn sind (siehe Kapitel 1), stellen sie wichtige Elemente sowohl von Business- als auch IT-Betriebsumgebungen dar. Eine angemessene Verwaltung ist somit unverzichtbar. In diesem Kapitel soll erläutert werden, wie diese Handhabung aussehen sollte.

Was es heißt, eine API zu verwalten

Ein gut gemachtes Entwicklerportal ist für einen API-Nutzer das Wichtigste. Für einen API-Anbieter ist die Verwaltung der API-Offenlegungs- und Freigabeprozesse jedoch nur die Spitze des Eisbergs (siehe Abbildung 2-1). Unter der Wasseroberfläche befinden sich die Business- und IT-Interessen, deretwegen Erstellung, Implementierung und Einsatz von APIs so zweckmäßig ist. Zu diesen Interessen gehören Datenzuordnung, Sicherheit, Geschwindigkeitsdrosselung, Überwachung und Versionsverwaltung.

Eine verwaltete API verfügt nicht nur über eine klar definierte Schnittstelle und eine definierte Zielgruppe; für sie gelten außerdem entsprechend durchgesetzte Business- und IT-Steuerungsmechanismen. Wie Sie in diesem Abschnitt sehen werden, gelten für verschiedene Gruppen unterschiedliche Anforderungen im Hinblick auf die API-Verwaltung.



Abbildung 2-1: Die Verwaltung von APIs erfordert mehr als API-Design und -Offenlegung.



Um maximale Effektivität zu erreichen, muss die Benutzererfahrung für alle drei Funktionen, die in diesem Kapitel angesprochen werden – Business Owner, IT Operations und API-Designer – individuell gestaltet sein. Ihre Interessen und Anliegen unterscheiden sich grundsätzlich, sodass die Werkzeuge einer Funktion für eine andere Funktion nicht geeignet sind.

API-Verantwortliche im Business

Der API-Verantwortliche im Business entscheidet Folgendes:

- ✓ Die Pläne (die Bedingungen) für die Nutzung der API
- ✓ Die Communitys, mit denen die API geteilt werden soll
- ✓ Ob die API ihre Ziele erreicht (andernfalls muss das Geschäftsmodell angepasst werden)

All dies kann erfolgen, ohne dass die API-Definition oder -Implementierung in irgendeiner Weise geändert wird. Weitere Informationen zur Funktion des Verantwortlichen im Business finden Sie unter „Warum API-Governance nötig ist“ weiter unten in diesem Kapitel.

IT-Betrieb

Der IT-Betrieb muss bestimmte, den Einsatz der API betreffende Merkmale sicherstellen, was ebenfalls ohne irgendeine Änderung

der API-Definition oder -Implementierung erfolgen kann. Es handelt sich hierbei um folgende Merkmale:

- ✓ Die Laufzeitumgebung, die die API hostet, kann sicher und zuverlässig betrieben werden.
- ✓ Die API wird ordnungsgemäß authentifiziert und alle Benutzer der API verfügen über adäquate Berechtigungen.
- ✓ Der API-Datenverkehr wird im Hinblick auf geschäftliche Anforderungen optimiert und priorisiert.



Zwischen dem, was der API-Verantwortliche an API-Planzugriffsmengen verkauft, und dem, was die zugrunde liegende IT-Infrastruktur bereitstellen kann, besteht ein wesentlicher Unterschied. Das Vorhalten ungenutzter Kapazität zur Unterstützung des gesamten potenziellen Datenverkehrs, der den verkauften API-Plänen entspricht, kann sehr kostspielig sein.



Zur Vermeidung untragbarer Laufzeitkosten sollten Sie sicherstellen, dass Ihre API-Laufzeitumgebung hoch skalierbar ist (sodass die eigentliche Last nicht so ins Gewicht fällt), oder den Datenverkehr drosseln, wenn die aktuelle Last die verfügbare Kapazität überschreitet, um Übertragungsspitzen zu glätten. Diese Möglichkeiten sollten in der ausgewählten API-Laufzeitumgebung verfügbar sein.

Weitere Informationen zur Funktion des IT-Betriebs finden Sie unter „Warum API-Governance nötig ist“ weiter unten in diesem Kapitel.

API-Designer

Der API-Designer ist für die physische Erstellung und Bereitstellung der API zuständig. Er muss Folgendes tun:

- ✓ Die API-Schnittstelle definieren
- ✓ Back-End-Endpunkte ermitteln, die die Daten oder Funktionen bereitstellen können, die für die Implementierung der API erforderlich sind
- ✓ Die Zuordnung zwischen der API-Schnittstelle und den Back-End-Daten- oder Funktionsquellen konfigurieren



Der API-Designer muss diese Aufgaben ohne großen Programmieraufwand erledigen können. Sobald die Erstellung einer API ein codelastiges Unterfangen anstatt eine Frage dynamischer Konfigurationen wird, nimmt das Innovationstempo auch bei den agilsten Entwicklungsteams unweigerlich ab. Die Unterscheidung zwischen der Konfiguration einer API und der Entwicklung der Back-End-Daten oder Funktion, die dieser API eine konkrete Form geben, ist elementar für eine API-Strategie. Wie in Kapitel 1 bereits erwähnt wurde, sind moderne APIs keine

Softwareressourcen, sondern eine flexible Möglichkeit, um Funktionalität für Zielgruppen jenseits des eigenen Teams darzustellen.

Warum API-Governance notwendig ist

Ein verbreiteter Mythos in API-Kreisen besagt, dass durch Governance alles ins Stocken gerät. Governance dient jedoch dazu, richtige Entscheidungen zu treffen. Governance bedeutet, sicherzustellen, dass die richtigen Personen zum richtigen Zeitpunkt aus den richtigen Gründen auf der Grundlage der richtigen Informationen die richtigen Entscheidungen treffen. Wenn eine API für Ihre Organisation wichtig ist, empfiehlt es sich daher, bei den diesbezüglichen Entscheidungen große Sorgfalt walten zu lassen. Als API-Anbieter oder API-Nutzer müssen Sie zahlreiche Entscheidungen zu APIs treffen.



Diese Governance unterscheidet sich von der Art von Governance, die üblicherweise im Rahmen des Softwarebereitstellungszyklus Anwendung findet. Sie ist jedoch nicht weniger wichtig.

Entscheidungen von API-Anbietern

Die Entscheidung, wer eine API unter welchen Geschäftsbedingungen nutzen kann, ist Sache des API-Verantwortlichen. Diese operative Business-Entscheidung gilt für alle APIs, unabhängig davon, ob es sich um die APIs handelt, die Ihr Entwicklerteam nutzt, um mobile Apps zu entwickeln, die APIs, die Sie für die geschäftsfeldübergreifende Integration verschiedener Systeme verwenden, oder die APIs, die von externen Nutzern verwendet werden. Es empfiehlt sich daher, für diese verschiedenen Zielgruppen jeweils spezifische Entscheidungen im Hinblick darauf zu treffen, welche APIs die jeweilige Zielgruppe unter welchen Bedingungen verwenden kann.

Der IT-Betrieb muss darüber hinaus in der Funktion des API-Anbieters geeignete Entscheidungen treffen- üblicherweise in Form von Sicherheits- und Datenverkehrsrichtlinien-, um die Infrastruktur vor Missbrauch oder Überlastung zu schützen.

Die Governance-Steuerung muss unkompliziert und die Entscheidungen müssen im Gegensatz zu den üblichen Lebenszyklusentscheidungen, die während eines Softwarebereitstellungszyklus getroffen werden, operativer Natur sein. Wenn es nicht ohne Weiteres möglich ist, die richtigen Entscheidungen zu treffen und durchzusetzen, beschädigt dies den offenen und dynamischen Charakter von APIs (wie Sie sich sicherlich erinnern, sind gute API-Implementierungen mit Konfigurationen und nicht mit Code umzusetzen). Zu einer guten API-Verwaltung gehören Business- als auch IT-Entscheidungen, die

daher von der gewählten API-Plattform unterstützt werden sollten. (Weitere Informationen zu API-Middleware finden Sie in Kapitel 5.)

Entscheidungen von API-Nutzern

Auch die Nutzer von APIs müssen wohl überlegte Entscheidungen treffen. Sie müssen insbesondere entscheiden, welche APIs sie für welche Zwecke verwenden möchten, und dann die folgenden Fragen zu jeder API stellen:

- ✓ Wie sieht das Bezahlmodell für die Verwendung der API aus und ist es für Ihre Zwecke angemessen?
- ✓ Benötigen Sie einen der API vorgelagerten Unternehmensproxy, um Lizenzen, Bezahlung u.ä. zu organisieren, oder registriert sich jeder Entwickler separat?
- ✓ Ist die API sicher und zuverlässig, um für unternehmenskritische Zwecke eingesetzt werden zu können? Alle protokollierten Aufzeichnungen zum langfristigen Verhalten der API können dazu beitragen, das Vertrauen der Nutzer im Hinblick auf die Verwendung der API zu stärken.

Wenn es sich bei den genutzten APIs um eigene APIs handelt, lassen sich diese Entscheidungen ziemlich leicht treffen, da es in diesem Fall vorwiegend um das allgemeine Business-Design geht. Falls es sich um APIs von Drittanbietern handelt, ist die Entscheidung schwieriger. Die Endbenutzererfahrung und die Verantwortung für die Aufrechterhaltung der Business-Integrität kann letztlich nicht delegiert werden. Sie benötigen eine Person innerhalb der eigenen Organisation, die für die Endbenutzererfahrung zuständig ist und die richtigen Entscheidungen im Hinblick darauf trifft, bei welchen APIs die Nutzung im Rahmen des eigenen Bereitstellungsmodells zweckmäßig ist.

Ein Plädoyer für nicht verwaltete APIs

Jeder weiß, dass moderne APIs durch eine API-Verwaltungslösung unterstützt werden müssen, oder? Immer schön langsam! Nicht alle APIs sind notwendigerweise verwaltete APIs. Im gesamten Kapitel wurde gezeigt, was gute API-Verwaltung ist. Was bedeutet es jedoch für eine API, wenn sie nicht verwaltet ist?

Im Folgenden sind die wichtigsten Unterschiede zwischen verwalteten und nicht verwalteten APIs aufgeführt:

- ✓ Eine nicht verwaltete API kann eine zukünftige Zielgruppe haben; diese Zielgruppe wird jedoch in den seltensten Fällen präzise definiert, geschweige denn durchgesetzt. Falls ein

Benutzer über Netzwerkzugriff auf die API verfügt, kann er sie normalerweise auch aufrufen.

- ✓ Für nicht verwaltete APIs gibt es keine unabhängig durchgesetzten Business- und IT-Steuerungsmechanismen. Jeder Steuerungsmechanismus wird mittels Logik in den API-Implementierungen, üblicherweise in Form von Code, bereitgestellt.

Mit anderen Worten: nicht verwaltete APIs verfügen weiterhin über eine klar definierte Schnittstelle, es gibt jedoch keine Möglichkeit, Kontrollmechanismen für ihr Laufzeitverhalten oder gar die Nutzer dieser API durchzuführen. Was spricht also überhaupt dafür, sich für eine nicht verwaltete API zu entscheiden?

Wenn eine API direkter Bestandteil Ihres Geschäftsmodells ist, wollen Sie wahrscheinlich nicht, dass sie nicht verwaltet ist. Gleichwohl gibt es Situationen, in denen die Verwendung einer nicht verwalteten API zweckmäßig oder sogar unvermeidlich ist. Hier ein paar Beispiele:

- ✓ Ein Gerät oder ein Sensor verfügt über eine definierte API als Teil seiner physischen Umsetzung, beispielsweise ein Hausthermostat, das per Fernzugriff programmiert werden kann, oder ein Fitbit (ein Wearable, das körperliche Aktivität überwacht), das Daten über eine definierte Schnittstelle mit einem Computer synchronisiert.
- ✓ Ein vorhandenes Softwaresystem, beispielsweise ein Standardsystem wie SAP oder ein Mainframesystem mit einer nativen REST-Schnittstelle, legt eine Mikro-API offen.
- ✓ Die API befindet sich innerhalb Ihrer eigenen Domäne und alles, was Sie für den Zugriff benötigen, ist Konnektivität.

Nicht verwaltete APIs können wichtige Ressourcen in vielen Ökosystemen sein, indem sie Schlüsselfunktionen und -daten auf einheitliche Weise verfügbar machen. Das einheitliche Nutzungsmodell ist der Grund, warum diese Schnittstellen weiterhin als APIs betrachtet werden sollten. Häufig kann es sich sogar empfehlen, alle nicht verwalteten APIs, die Ihnen zur Verfügung stehen, zu katalogisieren, sodass sie leicht zu finden sind und ihre Nutzung in einem bestimmten Programmiermodell so einfach wie möglich ist.

Aus allem eine API zu machen ist aus Nutzersicht das einfachste und effektivste Verfahren für Innovation und Zusammenarbeit in einer hybriden Umgebung. Das bedeutet, dass es bei einer API-Strategie nicht nur um die API-Verwaltung und verwaltete APIs gehen kann. Eine API-Strategie sollte Teil einer umfassenderen Integrationsstrategie sein, mit der aus Ihrem Unternehmen ein Motor für Innovationen wird.

Kapitel 3

Die Beschaffenheit guter APIs

In diesem Kapitel

- ▶ Angepasste APIs verwenden
- ▶ APIs mit serviceorientierter Architektur kombinieren
- ▶ Gutes API-Design erkennen

Schnelle Innovation wird durch gutes Design ermöglicht, was – egal über welche API wir sprechen – die Schnittstelle und die technischen Merkmale der API einschließt. Gutes Design beinhaltet aber auch die Entscheidung, welche APIs bereitgestellt werden und wann dies der Fall sein soll. Es gibt viele verschiedene API-Arten und Einsatzmöglichkeiten. In diesem Kapitel erfahren Sie, was eine gute API ausmacht.

APIs im Vergleich mit Rennwagen

Die Entwicklung und Weiterentwicklung eines Formel-1-Rennwagens durch ein Team liefert ein anschauliches Bild für das Wesen einer API (siehe Abbildung 3-1). In der Formel 1 ist jeder Rennwagen, der jemals an einem Rennen teilgenommen hat, ein Prototyp. Kein Team nimmt mit dem gleichen Wagen an zwei aufeinanderfolgenden Rennen teil. Ein Rennwagen besteht aus schnell austauschbaren Komponenten mit klar definierten Schnittstellen; der Wagen selbst ist mit integrierten Steuerungsmechanismen und Analysetechnologien ausgestattet. Auch wenn Teile des Wagens für die Dauer der gesamten Saison gleich bleiben, werden andere Komponenten kontinuierlich auf der Basis der Erkenntnisse optimiert, die im vorherigen Rennen gewonnen wurden.

Moderne Unternehmen sind in vielerlei Hinsicht wie Formel-1-Teams. Sie versuchen kontinuierlich, ihr Geschäftsmodell zu verbessern und das richtige Gleichgewicht zwischen Veränderung und Stabilität zu finden. APIs sind eine Möglichkeit, wie Experimentierfreude zum Nutzen des Unternehmens eingesetzt werden kann. „Früh testen, schnell lernen und ohne großen Aufwand erweitern“ ist ein Prinzip, das sich sehr gut eignet, um es auf die API-Welt anzuwenden.

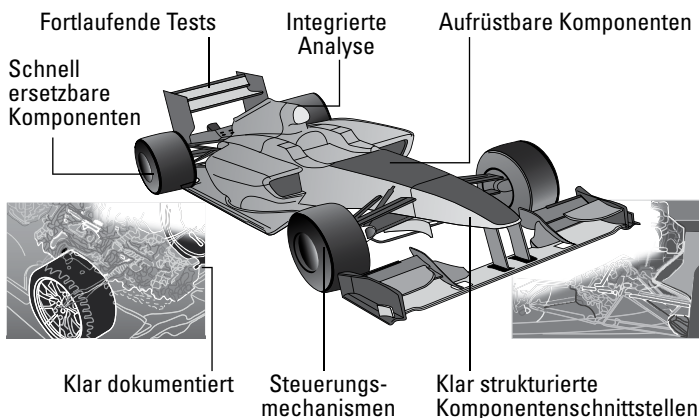


Abbildung 3-1: Die Entwicklung von Rennwagen und APIs sollte den gleichen Konstruktionsprinzipien folgen.

Ein Plädoyer für angepasste APIs

Müssen APIs stets so konzipiert sein, dass sie wiederverwendet werden können? Wiederverwendbarkeit impliziert Stabilität über einen relativ langen Zeitraum und Stabilität ist sinnvoll, wenn eine API für die Partnerintegration verwendet oder extern als Teil Ihres Geschäftsmodells zugänglich gemacht werden soll. Wenn die API jedoch lediglich erstellt wurde, um beispielsweise die Zusammenarbeit zwischen einem Team für mobile Entwicklung und dem Team zu verbessern, das für die Pflege eines Back-End-Systems zuständig ist, ist Wiederverwendbarkeit möglicherweise weder erwünscht noch zweckmäßig.

Die Verwendung einer API muss attraktiv sein und für einen Entwickler muss die Verwendung dieser API schneller und zweckdienlicher sein als die Programmierung einer anderen Lösung. Wenn sich die Anforderungen eines Entwicklers mobiler Lösungen schnell ändern, müssen sich die APIs genauso schnell ändern, um Schritt zu halten. Damit hätten wir das Argument für *angepasste APIs* – schnell erstellte, sich schnell ändernde APIs, die darauf ausgelegt sind, eine bestimmte Nutzeranforderung zu erfüllen.



Wiederverwendbarkeit oder Stabilität über einen längeren Zeitraum sind keine Eigenschaften, die notwendiger zum Konzept einer API gehören. Welche Bedeutung Wiederverwendbarkeit und Stabilität haben, hängt allein von dem Geschäftszweck ab.

Die Bereitstellung angepasster APIs setzt voraus, dass die Erstellung und Pflege von APIs sowohl einfach als auch kostengünstig sind. Andernfalls würden Anpassungen zu unangemessen hohen Kosten führen.

Gute API-Verwaltungslösungen erstellen APIs mittels Konfiguration anstatt per Programmierung und die Erstellung und Änderung einer API dauert normalerweise nur wenige Minuten. Ungeachtet der Kosten für Erstellung und Pflege ist die ordnungsgemäße Verwaltung auch angepasster APIs von großem Nutzen.



Die Verwaltung angepasster APIs bietet die folgenden Vorteile:

- ✓ Definition und Durchsetzung von Business- und IT-Steuerungsmechanismen
- ✓ Allgemeine Erkenntnisse zur Leistungsfähigkeit einer API aus Geschäftsperspektive
- ✓ Flexibilität des IT-Betriebs, um API-Workloads zu verschieben und dynamisch anzupassen

Diese Vorteile sind wichtige Aspekte des Prinzips „früh testen, schnell lernen und ohne großen Aufwand erweitern“ und enorm wichtig, um Wandel in einer Welt voller Chancen und Innovationen zu optimieren.

Ein paar Überlegungen zu APIs und Services

Seit mehr als zehn Jahren ist die Idee serviceorientierter Architekturen (SOA) ein etabliertes Konzept; der moderne API-Begriff ist deutlich jünger. Beide Integrationsansätze haben ihre Anhänger und gehen auf Business- wie auch auf IT-Fragen ein. Worin besteht also der eigentliche Unterschied zwischen diesen beiden Ansätzen und ist es nötig, sich für einen zu entscheiden?

APIs im Vergleich zu Services

Das Kernkonzept von SOA ist die Service-Idee. So definiert beispielsweise The Open Group einen Service als „eine logische Darstellung einer wiederholbaren Aktivität, die zu einem bestimmten Ergebnis führt“. Services sind eigenständig und für ihre Nutzer nicht transparent; darüber hinaus verfügen sie über klar definierte Interaktionsvereinbarungen. Aus technischer Sicht gelten diese Merkmale auch für jede klar definierte API; somit *ist* eine API technisch gesehen ein Service.

Ist API also einfach nur eine andere Bezeichnung für Service? Zwischen Services und APIs gibt es einen bedeutenden Unterschied- das ist die Zielrichtung ihres Designs (siehe Abbildung 3-2). APIs sind so konzipiert, dass sie für den zukünftigen Nutzer interessant sind; sie ändern sich, sobald sich die Anforderungen des Nutzers ändern. Demgegenüber stehen bei der Entwicklung von Services die allgemeinen Kosten und die Stabilität im Vordergrund.

„Wie kann ich das **Innovationstempo** erhöhen?“

APIs



„Wie kann ich die **Flexibilität und Effektivität der Bereitstellung** erhöhen?“

Serviceorientierte Architektur (SOA)

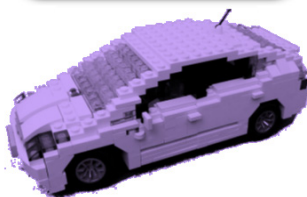


Abbildung 3-2: APIs und Services greifen unterschiedliche Fragestellungen auf.

Für einen Nutzer stehen bei der Verwendung einer API die Aspekte Schnelligkeit, Zweckmäßigkeit und geringer Lernaufwand im Vordergrund. Diese Designkriterien sind die grundlegenden Unterschiede zwischen APIs und der klassischen Vorstellung von Services:

- Für einen Serviceanbieter steht beim Thema „Wiederverwendung“ der Aufwand für die Bereitstellung einer API im Mittelpunkt. Für API-Nutzer geht es beim Thema „Wiederverwendung“ um die Geschwindigkeit, mit der sie ihre Software bereitstellen können, und zwar ungeachtet der Kosten für die Bereitstellung der APIs, die als Teil dieser Software genutzt werden.
- Für einen Serviceanbieter steht beim Thema „gemeinsame Nutzung“ die Effektivität im Vordergrund. Für einen API-Nutzer geht es bei der gemeinsamen Nutzung um Zweckmäßigkeit (wenn sie nicht zweckmäßig ist, wird die API nicht verwendet).
- Für einen Serviceanbieter geht es beim Thema „Kapselung“ darum, nicht viel *ändern* zu müssen. Für einen API-Nutzer bedeutet „Kapselung“, nicht viel *lernen* zu müssen (wenn die Schnittstelle komplex ist, wird die API nicht verwendet).

Auf der einen Seite möchte ein Entwickler mobiler Lösungen, dass die Schnittstelle im Hinblick auf seine konkrete App möglichst einfach ist. Demgegenüber steht das Back-End-Team, dem daran gelegen ist, dass jeder den gleichen standardisierten Service und das gleiche Datenmodell verwendet. Gibt es, anstatt eine Lösung dieses Konflikts zu erzwingen, eine Möglichkeit, beiden Seiten gerecht zu werden, ohne unerschwingliche Kosten in Kauf nehmen zu müssen?



Eine historische Analogie lässt sich bei der Entwicklung von Datenbanken feststellen. Die ersten Datenbankgenerationen waren ausschließlich auf interne Faktoren, also die Tabellen und Schemas, ausgerichtet. Nach kurzer Zeit zeigte sich jedoch bereits die Notwendigkeit, kontrollierte Teilmengen der Daten in bestimmter Form und auf

eine bestimmte Gruppe von Datennutzern zugeschnitten zugänglich zu machen – woraus sich die Idee einer Datensicht als Kernfunktionalität der meisten modernen Datenbanken entwickelte: ein schlanker Proxy oberhalb des Wertebereichs, der durch die internen Objekte dargestellt wird.

APIs sind kontrollierte (Proxy-) Sichten der Daten und Funktionalität einer Domäne, die für die Anforderungen von API-Nutzern optimiert sind. Solange das Erstellen und Pflegen von Proxy-APIs preiswert ist, können Sie mit ihrer Hilfe eine Domäne auf unterschiedliche Weise darstellen und dadurch für jede Gruppe von API-Nutzern optimieren. (Sie möchten Ihren externen Partnern doch sicherlich eine andere Darstellung Ihrer Funktionalität als Ihren internen Entwicklern liefern.)

APIs und Services im Team

SOA wurde „erfunden“, um Servicenutzer von Änderungen am Backend abzusichern. Aber wer schützt die Serviceanbieter vor den Unwägbarkeiten sich ändernder Anforderungen in Omnichannel-Front-End-Lösungen? Durch die Kombination von APIs und Services können Sie ein Zentrum der Ruhe inmitten des Sturms der Veränderung schaffen. Services sind das Instrument, mit dem Anbieter die Basisfunktionalität ihrer Domänen festschreiben. APIs sind der Mechanismus, mit dem diese Funktionalität (Services) neu verpackt, zu einem Produkt gemacht und dann in einer benutzerfreundlichen Form mit anderen geteilt wird. APIs und Services ergänzen sich eher, als dass sie sich widersprechen, und können, wenn sie gemeinsam angewendet werden, die Innovationsfähigkeit eines Unternehmens dramatisch erhöhen.

Gutes API-Design erkennen

Das technische Design von APIs ist zweifelsohne wichtig, aber es variiert erheblich in Abhängigkeit von der Technologie, die für das Design und die Implementierung einer speziellen API ausgewählt wird. Über das Design von Schnittstellen wurden ganze Bücher geschrieben, die dieses Thema in einer Ausführlichkeit behandeln, die weit über das hinausgeht, was dieses Büchlein leisten kann. Hier soll der Hinweis genügen, dass das Schnittstellendesign von APIs im Allgemeinen gut bekannt ist, wie die folgenden Beispiele zeigen:

- REST-Schnittstellen sind ressourcenbasiert. Der wichtigste Aspekt des Schnittstellendesigns ist die URI-Struktur, die einem Nutzer die Navigation in dem von der API verkörperten Objektdiagramm ermöglicht.
- SOAP-Schnittstellen sind methodenbasiert. Die wichtigsten Aspekte des Schnittstellendesigns sind die Menge an unterstützten Methoden und die Datenstrukturen jeder Methode.

- ✓ MQTT-Schnittstellen sind ereignisbasiert. Die wichtigsten Aspekte des Schnittstellendesigns sind die Menge der Ereignisse (ausgegeben oder empfangen) und die zugehörigen Ereignisnachrichten.

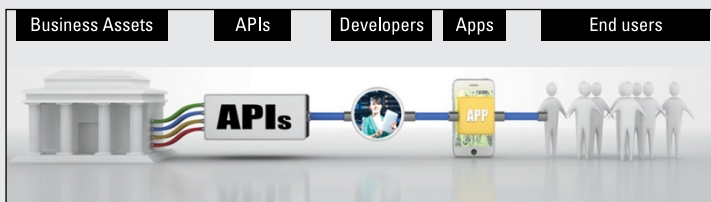


Nicht alle APIs sind REST-APIs. REST-Schnittstellen eignen sich in der Regel sehr gut für die Nutzung durch Menschen und sind derzeit bei Entwicklern mobiler Lösungen ausgesprochen beliebt. REST-Schnittstellen neigen jedoch dazu, geschwätzig zu sein. Und obwohl sie erweiterbar sind, transportieren sie keine stark typisierten komplexen Datenstrukturen. SOAP-Schnittstellen eignen sich hervorragend für die Integration zwischen Systemen und werden von IT Operations-Teams wegen ihrer weniger stark ausgeprägten Geschwätzigkeit und der präziseren Datenstrukturen bevorzugt. MQTT-Schnittstellen werden für die Kommunikation mit dem Internet der Dinge bevorzugt, in welcher Bandbreite und Akkulaufzeit die größten Sorgen sind und in der garantierte Nachrichtenzustellung zwischen der Vermeidung von Störfällen und ihrer unabsichtlichen Inkaufnahme unterscheiden kann.

Der Wert von APIs ist mehr als die so genannte API-Wertschöpfungskette

Ein großer Teil des API-Hypes dreht sich um die API-Wertschöpfungskette, die die Abbildung in dieser Zwischenbemerkung veranschaulicht – die Art

und Weise, wie APIs ein Geschäftsmodell zu einem offenen Ökosystem erweitern.



Leider geht es in den meisten Beispielen, die in der Branche lang und breit diskutiert werden, ausschließlich um öffentliche APIs, was jedoch längst nicht der einzige Anwendungsfall für APIs ist. Was jedoch noch schwerer wiegt: Diese allgemeine Darstellung der Wertschöpfungskette ist im Hinblick auf die Frage, welche APIs warum bereitgestellt werden sollten,

uninteressant. Der Grund hierfür ist, dass diese Darstellung nichts über die verschiedenen Arten von Geschäftszielen aussagt, die dazu geführt haben, dass eine API überhaupt in Betracht gezogen wurde. Kapitel 4 befasst sich mit dieser konzeptionellen Lücke, indem die typischen API-Einstiegs- und Abstiegspunkte mit den zugehörigen Entscheidungskriterien definiert werden.

Kapitel 4

API-Einstiegspunkte

In diesem Kapitel

- ▶ Daten monetarisieren
- ▶ Innovationen nutzen
- ▶ Die Umstellung auf Mobile beschleunigen
- ▶ Jetzt wird's hybrid
- ▶ Alles wird programmiert

Was bedeutet es wirklich, eine API-Strategie zu haben? Der ein oder andere wird hier einfach auf die so genannte API-Wertschöpfungskette (siehe Kapitel 3) verweisen, aber ist das wirklich alles? Deshalb die ganze Aufregung? Ich bin überzeugt, dass es für eine API-Strategie mehr braucht. Wie in Kapitel 3 erläutert wurde, müssen Entscheidungen im Hinblick auf das richtige Gleichgewicht zwischen Unternehmens- und angepassten APIs getroffen werden. Hierzu gehören Entscheidungen in Bezug auf die Bedingungen, unter denen APIs geteilt werden können, und Entscheidungen hinsichtlich der Frage, wie erforderliche APIs während der API-Implementierung zu vorhandenen Assets zugeordnet werden sollen. Diese und viele weitere Entscheidungen hängen von dem gewünschten Ergebnis für das Business ab.

Das zentrale Element des Begriffs „API-Strategie“ ist das Wort „Strategie“. Überlegen Sie, was Sie aus geschäftlicher Sicht erreichen wollen, welche Zielgruppe angesprochen werden soll, welche Arten von APIs erforderlich sind, um die Zielgruppe zu erreichen, und wie Sie Ihre Daten- und Anwendungs-Assets (als Services) zusammenstellen, um die APIs zu unterstützen, die Sie bereitstellen müssen. Vergessen Sie dabei nicht, darüber nachzudenken, welche APIs Sie selbst nutzen möchten und von wem diese APIs stammen. Eine API-Strategie betrifft nicht nur API-Anbieter; viele Organisationen nutzen ein Vielfaches der Anzahl an APIs, die sie selbst bereitstellen. Diese Fragen sind die zentralen Elemente einer erfolgreichen API-Strategie.

In diesem Kapitel werden fünf API-Einstiegspunkte definiert, die nach meiner Erfahrung charakteristisch für die Business- und

IT-Interessen sind, die eine API-Strategie maßgeblich bestimmen. Ein Unternehmen kann verschiedene Absichten gleichzeitig verfolgen, aber jede dieser Absichten führt zu anderen Entscheidungskriterien, was den Einsatz von APIs betrifft. Sie können dieses Kapitel von Anfang bis Ende lesen oder auch direkt zu dem Einstiegspunkt springen, der am besten zu Ihren geschäftlichen Anforderungen passt.

Daten monetarisieren

Die Monetarisierung Ihrer Daten basiert auf der Offenlegung von Erkenntnissen oder Funktionen in einer Form, die Dritte dazu bewegt, diese Erkenntnisse oder Funktionen zu verwenden. Die Monetarisierung kann in unterschiedlichster Form erfolgen. Das offensichtlichste Beispiel ist, wenn Dritte für die Nutzung Ihrer API bezahlen. In anderen Fällen sind möglicherweise Sie es, der den API-Nutzer als Gegenleistung für eine größere geschäftliche Reichweite und ein stärkeres Ökosystem bezahlt. Oder Sie setzen APIs ein, um Partnern den Einstieg zu ermöglichen, ohne dass überhaupt ein direkter Geldfluss damit verbunden ist. Die zentralen Geschäftsziele bei der Monetarisierung sind eine Neuausrichtung des Business, eine Veränderung der Wertschöpfungskette und die Vergrößerung von Reichweite und Einfluss.

Erfolg mit diesem Einstiegspunkt setzt in der Regel eine sorgfältige Planung voraus. Obwohl Sie bis zu einem gewissen Grad experimentieren können und sollten, muss das endgültige Produkt aus einer Menge stabiler Unternehmens-APIs bestehen, auf die sich Dritte für einen langen Zeitraum verlassen können.

Hier ein paar Empfehlungen für eine API-Strategie mit dem Fokus auf der Monetarisierung Ihrer Daten:

- ✓ **Ziel:** Das gewünschte Ergebnis ist finanzieller Art oder basiert auf einem nicht monetären Wert wie der Ausweitung Ihres Einflussbereichs.
- ✓ **Zielgruppe:** Die Zielgruppe ist zwangsläufig eine Drittpartei. Typische Fälle sind Partner und externe Entwickler (keine Entwickler Ihrer eigenen Organisation oder Entwickler, die von Ihrer eigenen Organisation engagiert wurden). Es ist nicht ungewöhnlich, einen anderen Geschäftsbereich innerhalb Ihres Unternehmens als Partner oder Drittpartei zu behandeln, insbesondere dann, wenn verschiedene Geschäftssparten als wirtschaftlich eigenständige Gebilde behandelt werden.
- ✓ **Bereitzustellende APIs:** Die APIs, die erforderlich sind, um die Zielgruppe anzusprechen, müssen den Wert liefern, den Sie verkaufen möchten. Diese Auswahl will wohl überlegt sein, und zwar nicht nur im Hinblick auf den eigentlichen Wert, der

bereitgestellt wird, sondern auch hinsichtlich der Form, die die Nutzung attraktiv macht.

- ✓ **API-Nutzungsbedingungen:** Vergessen Sie bei Ihren Überlegungen nicht die Bedingungen, die die Nutzung der API regeln, beispielsweise Freemium, nutzungsabhängige Bezahlung oder Vorauszahlung.
- ✓ **API-Implementierung:** Beim Zusammenstellen von Daten und Funktionen für die API-Implementierung kommt es auf Qualität und Zuverlässigkeit an. Es gibt Menschen, die behaupten, dass die Kosten der Implementierung der wichtigste Faktor sind, aber eine Strategie zur Monetarisierung von Daten steht und fällt nicht mit den Implementierungskosten. Viel entscheidender für die langfristige Tragfähigkeit ist die Frage, ob die anvisierten API-Nutzer sowohl Nutzen als auch Zuverlässigkeit erfahren.
- ✓ **Nutzung fremder APIs:** In einigen Fällen müssen Sie außerdem berücksichtigen, welche APIs Sie selbst nutzen müssen. Auch wenn der von Ihnen generierte Wert in erster Linie über die Bereitstellung von APIs zur Fremdnutzung entsteht, kann die Implementierung dieser APIs die Erstellung übergeordneter Zusammensetzungen bereits vorhandener APIs beinhalten. Dies geschieht in der Regel dadurch, dass diese vorhandenen APIs mit etwas kombiniert werden, das eindeutig von Ihnen stammt.

Der Einstiegspunkt „Daten monetarisieren,“ ist vermutlich derjenige, von dem Sie am häufigsten gehört haben, aber er ist nicht der gebräuchlichste. Derzeit bezieht sich eine signifikante Mehrheit der API-Initiativen auf interne Anwendungsfälle, was möglicherweise auch dann so bleiben wird, wenn der Austausch über öffentliche APIs völlig ausgereift ist.

Innovationsfreiheit

Für viele Unternehmen ist Innovationsfreiheit das Gebot der Stunde. „Früh testen, schnell lernen und ohne großen Aufwand erweitern“, dieses Prinzip beschreibt die Schlüsselmerkmale eines dynamischen, attraktiven Unternehmens. Der Schwerpunkt dieses Einstiegspunkts liegt auf der aggressiven Verfolgung von Geschäftschancen und einer Haltung, die Innovationen als Lernprozess basierend auf folgendem Modell begreift:

- ✓ Alles gilt so lange als Prototyp, bis es sich in der Praxis bewährt hat.
- ✓ Festzustellen, dass etwas nicht funktioniert, ist kein Versagen, sondern normaler Bestandteil eines Lernprozesses.
- ✓ Stillstand führt unweigerlich zum Niedergang.

Wie bereits in Kapitel 3 ausgeführt, besteht die Rolle von APIs darin, das ruhige Zentrum im Sturm des Wandels zu bilden. Diese Rolle hat zwei Aspekte:

- ✓ Schnelle Bereitstellung dessen, was der experimentierende API-Nutzer benötigt, und Entfernen von Funktionalität, wenn sie nicht mehr benötigt wird
- ✓ Schutz des Anbieters vor unkontrollierbaren Veränderungen (in Kapitel 3 können Sie bei Bedarf noch einmal nachlesen, warum die Definition und Bereitstellung neuer API-„Sichten“ auf bestehende Assets einfach und kostengünstig sein sollte)



Der Einstiegspunkt „Innovationsfreiheit“ ist möglicherweise nicht so glamourös wie die Monetarisierung von Daten (siehe den vorherigen Abschnitt), es ist jedoch der API-Anwendungsfall, den IBM in den meisten Großunternehmen mit Abstand am häufigsten beobachtet. Die Fähigkeit, neue, innovative Funktionen, egal ob intern und/oder extern, zu entwickeln, ohne dass die Kosten explodieren, ist eine Herausforderung, mit der jedes Unternehmen konfrontiert ist. Zur Beschleunigung von Innovationen ist eine Mischung aus sorgfältiger Planung und angepasster Reaktion erforderlich.

Hier ein paar Empfehlungen für eine API-Strategie mit dem Fokus auf dem Aspekt der Innovationsfreiheit:



- ✓ **Ziel:** Das gewünschte Ergebnis ist die schnelle Erkenntnis, was funktioniert und was sich am Markt hervorhebt, um dann Erfolge weiter auszubauen.

Auch, wenn Sie glauben, die Marktbedürfnisse zu kennen, kann eine Haltung des Lernens ein hilfreicher Augenöffner sein. Bei den komplexen Marktvorgängen passiert es schnell, dass Symptom und Ursache miteinander vermengt werden.

- ✓ **Zielgruppe:** Die Zielgruppe umfasst vorwiegend interne Entwickler, egal, ob sie zu Ihrer Organisation gehören oder die Entwicklung outgesourct wurde. Mitunter werden externe Agenturen engagiert, um eigene Innovationsanstrengungen zu unterstützen. In diesen Fällen sind offiziellere Vereinbarungen erforderlich, die regeln, welche APIs wann bereitgestellt werden sollen. Doch auch dann empfiehlt es sich, einen gewissen Opportunismus an den Tag zu legen, um das erforderliche Tempo beizubehalten. Vertragsverhandlungen sind der Fluch schneller Innovation.

- ✓ **Bereitzustellende APIs:** Die APIs, die erforderlich sind, um die Zielgruppe anzusprechen, sind eine Mischung aus vordefinierten Unternehmens-APIs und angepassten APIs (siehe Kapitel 3), die aus den Anforderungen einer innovativen App

abgeleitet werden. Die Beimischung von Unternehmens-APIs, um zentrale Daten auf leicht zugängliche Art offenzulegen, kann nicht nur die Entwicklung, sondern auch die Ideenfindung auf Trab bringen.

Sorgen Sie dafür, dass die vordefinierten APIs klein und einfach bleiben. Würde die gesamte Datenstruktur eines Kunden-Back-End-Systems offengelegt, würde dies die Nutzung durch einen innovativen Channel-Entwickler nicht unbedingt einfacher machen.

- ✓ **API-Nutzungsbedingungen** Die Bedingungen der API-Nutzung bleiben auch hier wichtig, allerdings nicht unter dem Aspekt Bezahlung, sondern im Hinblick auf die Aufrechterhaltung der Sicherheit und Stabilität von Back-End-Systemen. Innovation ist schließlich nicht vorhersehbar.
- ✓ **API-Implementierung:** Das Zusammenstellen von Daten und Funktionen für die API-Implementierung variiert in Abhängigkeit davon, ob es sich um vorab geplante Unternehmens-APIs oder angepasste, bedarfsorientierte APIs handelt:

- *Bei vorab geplanten APIs* muss die Entscheidung, welche Datensegmente auf Organisationsebene offengelegt werden, gut überlegt sein. Darüber hinaus müssen bei der Implementierung (üblicherweise als Proxy für Unternehmensservices) die endgültigen Laufzeitkosten berücksichtigt werden. Vorab geplante APIs werden auf nicht vorhersehbare Weise verwendet; die Laufzeitkosten dürfen daher kein Faktor sein, der eine derartige Verwendung verhindert.
- *Bei angepassten APIs* sind Entwicklungsgeschwindigkeit und Entwicklungskosten die wichtigsten Aspekte, die zu berücksichtigen sind. Wenn Sie improvisieren müssen, um die API heute und nicht erste nächste Woche herausbringen zu können, können Sie dies tun, solange Sie einen tragfähigen Plan für die Bereinigung der API-Implementierung haben, falls diese API ein Erfolg wird.

Vielen angepassten APIs ist möglicherweise kein langes Leben beschieden. Falls es sich herausstellt, dass eine angepasste API Ihre aktuellen Anforderungen nicht mehr erfüllt, sollten Sie sie einfach entsorgen und den nächsten Durchgang Ihres Lernprozesses einleiten.

- ✓ **Nutzung fremder APIs:** Bei diesem Einstiegspunkt spielt die Frage, welche Drittanbieter-APIs genutzt werden sollen, eine wichtige Rolle. Als Beispiel sei hier die Erstellung von mobilen Social-Apps erwähnt, die ohne Zugriff auf APIs von bekannten,



öffentlichen sozialen Services wie Twitter, Facebook und LinkedIn ausgesprochen schwierig wäre. Diese Drittanbieter-APIs sollten zu dem API-Katalog gehören, den Sie Ihren internen Entwicklern zur Verfügung stellen. Es sollte nicht erforderlich sein, dass die Entwickler externe Sites aufsuchen müssen, um die benötigten APIs selbst aufzuspüren. Es kann sich sogar empfehlen, die Drittanbieter-APIs in einer eigenen, einfacheren Version zusammenzustellen, da einige der öffentlichen APIs in ihrer ursprünglichen Form ziemlich komplex sind.



Innovation ist niemals einfach, kann aber auf vielfältige Weise unterstützt werden. Durch die korrekte Verwendung von APIs kann das Unternehmens-Back-End zu einem wesentlichen Element Ihrer Innovationsbemühungen werden, anstatt diese zu behindern. Unternehmen mit langer Geschichte haben den Vorteil, dass sie über mehr Assets verfügen, die sie über APIs offenlegen können. Aber auch Startups bietet eine API-gestützte Herangehensweise bei der Implementierung innovativer Lösungen mehr Flexibilität im Hinblick auf die Beschaffung von Daten und Funktionen. Durch die Verwendung von APIs können sich Channel-Entwickler wieder mehr auf die Benutzererfahrung konzentrieren, anstatt sich um Integrationsfragen kümmern zu müssen. Durch die Verwendung von APIs kann außerdem die Omnichannel-Erfahrung unterstützt werden, da die Daten und Funktionen hinter der API per Definition Remoteressourcen sind und unternehmensweit über jeden Kanal oder jede Anwendung zugänglich sind.

Mobil in zehn Minuten

Dieser API-Einstiegspunkt hängt eng mit dem Einstiegspunkt „Innovationsfreiheit“ (siehe den vorherigen Abschnitt) zusammen und stellt in gewisser Hinsicht eine extreme Variante dieses Einstiegspunkts dar. Der Unterschied besteht darin, dass es bei diesem Einstiegspunkt ausschließlich darum geht, alles zu unterstützen, was Ihr Entwicklerteam für mobile Lösungen hier und jetzt benötigt.

Unternehmen müssen heute viele kleine, eigenständige Apps anstelle der herkömmlichen, groß angelegten Portale erstellen. Ich bin fest davon überzeugt, dass mobile Nutzer es vorziehen, ihre Omnichannel-Erfahrung selbst zu gestalten, anstatt sich mit dem zu begnügen, was jemand anderes für sie definiert hat. Ich kann es an mir selbst beobachten: Nur ganz selten verbringe ich mehr als ein paar Minuten in einer bestimmten mobilen App, bevor ich mich etwas anderem zuwende.

Der Einstiegspunkt „Mobil in zehn Minuten“ macht es notwendig, die Zusammenarbeit zwischen Back-End-Verantwortlichen und

Entwicklerteams für mobile Lösungen zu verbessern. Bei diesen Entwicklerteams kann es sich um interne oder externe Teams handeln, die jedoch in jedem Fall vorhandene Daten und Funktionen nutzen müssen, um eine attraktive Benutzererfahrung bereitzustellen. So mag beispielsweise die Möglichkeit, Aktienkurse anzuzeigen, bereits recht nützlich sein, aber erst die Möglichkeit, diese Aktienkursen im Kontext des eigenen Investmentportfolios zu nutzen, wäre ein echtes Unterscheidungsmerkmal.

Der Fokus dieses Einstiegspunkts ist kontrollierte Einfachheit. Verstecken Sie Komplexität, vereinfachen Sie, was der Entwickler mobiler Lösungen sieht und nutzt, und sorgen Sie gleichzeitig für geeignete Kontrollmechanismen für Business und IT. Achten Sie außerdem darauf, dass der Prozess wenig Zeit beansprucht, um mobile Innovationen nicht auszubremsen.



Einige IBM Entwickler haben zum Spaß getestet, ob es mithilfe eines API-Ansatzes möglich ist, ein Datenelement von einem Mainframesystem in 10 bis 15 Minuten auf einem mobilen Gerät bereitzustellen. Sie haben es tatsächlich geschafft! Es gab kein schickes API-Design, aber das Experiment zeigte, dass durch situationsgerechte Verwendung von APIs und Integrationstechnologien die Komplexität von Integrationslogik weitgehend außen vor bleiben kann.

Bei dem Einstiegspunkt „Mobil in zehn Minuten“ dreht sich alles um angepasste Innovationen. Hier ein paar Empfehlungen für eine API-Strategie in diesem Kontext:

- ✓ **Ziel:** Das gewünschte Ergebnis ist die unmittelbare Unterstützung der Anforderungen Ihrer Entwicklerteams mobiler Lösungen. Das Entwicklerteam ist dafür zuständig, die Daten zu bestimmen, die für die Benutzererfahrung notwendig sind, und die Back-End-Verantwortlichen erstellen die APIs, die exakt diese Daten liefern. Die Zuordnung zu vorhandenen Daten und Funktionen (Services) ist Teil der API-Implementierung und fällt nicht in die Verantwortung der Entwickler mobiler Lösungen.
- ✓ **Zielgruppe:** Die Zielgruppe für die APIs ist das Entwicklerteam mobiler Lösungen. Punkt!
- ✓ **Bereitzustellende APIs:** Die APIs, die erforderlich sind, um die gewünschte Zielgruppe zu erreichen, sind ausschließlich angepasste APIs. Vielleicht haben Sie das Glück, vorhandene APIs für eine neue mobile App einzusetzen; Sie sollten jedoch zu viele appübergreifende Abhängigkeiten von bestimmten API-Verträgen vermeiden.

- ✓ **API-Nutzungsbedingungen:** Die Bedingungen für die API-Nutzung dienen dazu, Sicherheit und Stabilität von Back-End-Systemen sicherzustellen. Gleiches gilt für den Punkt „Innovationsfreiheit“ (siehe den vorherigen Abschnitt).
- ✓ **API-Implementierung:** Da die erzeugten APIs sehr angepasst sind und nur selten einen längeren Zeitraum überdauern, sind Entwicklungsgeschwindigkeit und -kosten die wichtigsten Aspekte, die berücksichtigt werden müssen. Wenn Sie improvisieren müssen, um die API heute und nicht erste nächste Woche herausbringen zu können, können Sie dies tun, solange Sie einen tragfähigen Plan für die Bereinigung der API-Implementierung haben, falls diese API ein Erfolg wird.
- ✓ **Nutzung fremder APIs:** Um mobile Anwendungen attraktiv zu machen, ist in der Regel ein soziales Element erforderlich. Es ist daher unverzichtbar, sich Gedanken darüber zu machen, welche öffentlichen Social-APIs (wie Twitter, Facebook und LinkedIn) genutzt werden sollen und wie diese Nutzung kontrolliert wird. Es könnte sich anbieten, die Social-APIs anderer Anbieter in einer eigenen, einfacheren Version zusammenzustellen (siehe hierzu den Abschnitt „Innovationsfreiheit“ weiter oben in diesem Kapitel). Ganz bestimmt jedoch müssen Sie die Schattenseite von APIs (siehe Kapitel 6) berücksichtigen und versuchen, die Auswirkungen auf den Ruf Ihres Unternehmens zu minimieren, falls sich durch die Verwendung öffentlicher APIs Stabilitäts- oder ethische Probleme ergeben.

Der angepasste Ansatz dieses Einstiegspunkts kann den Eindruck erwecken, dass eine sinnvolle Verwaltung aufgrund der Anzahl der beteiligten APIs und ihrer notwendigerweise begrenzten Wiederverwendbarkeit nicht möglich ist. Tatsächlich wäre es allein mit der Technologie, die bis vor drei oder vier Jahren verfügbar war, für viele Großunternehmen praktisch unmöglich gewesen. Heute ist das jedoch anders! API-Verwaltung schafft Abhilfe, indem es problemlos eine schlanke nutzerorientierte API vom Portfolio softwarebasierter Unternehmensdaten und -services abgrenzt. Wenn die Erstellung einer neuen API kinderleicht und die Verwaltung einer großen Anzahl von APIs sowie ihre Freigabe für verschiedene Communitys fast ebenso einfach ist, sind Wiederverwendbarkeit und architektonische Genauigkeit von API-Schnittstellen nicht mehr ganz so wichtig. Stattdessen rückt die Frage in den Mittelpunkt, was für den Erfolg des API-Nutzers notwendig ist.

Leben in einer hybriden Welt

Kein Buch über APIs wäre vollständig, ohne das Verhältnis von APIs zur Cloud zu thematisieren. Der vierte API-Einstiegspunkt konzentriert sich auf die Verwendung von APIs als einheitliches Nutzungsmodell in einem hybriden Ökosystem aus lokalen Systemen und privaten und öffentlichen Cloudumgebungen. Das Business-Mantra ist „Entscheidungsfreiheit“ –frei zu entscheiden, wie beliebige Funktionen oder Daten bezogen werden sollen, und beliebig Lösungen für jeden gewünschten Formfaktor bereitzustellen.

Ein API-zentrierter Integrationsansatz wird durch Folgendes gefördert:

- ✓ Ein Unternehmen, das abgekoppelt ist, ist nicht wettbewerbsfähig. Die Cloud ist daher dasselbe wie hybride Lösungen aus Cloud- und lokal implementierten Teilen, sodass die Integration dementsprechend organisiert und gesteuert werden muss.
- ✓ Bei der Cloud geht es um Möglichkeiten (Business und IT) und nicht um Standorte; jedes übergreifende Nutzungsmodell muss daher standortunabhängig sein.
- ✓ In einer Welt, die durch ein „System aus Systemen“ definiert ist, gibt es keine herkömmliche Netzwerkperipherie mehr, der Geltung verschafft werden müsste; Interaktionen müssen daher auf Anwendungsebene gesteuert werden.

Wenn für einen Nutzer alles eine (remotefähige) API ist, muss der Nutzer nicht wissen, wo und wie diese API gehostet wird. Syndizierte API-Kataloge können und sollten für Transparenz über Domänen- und Anbietergrenzen hinweg sorgen. Eines der wichtigsten Hilfsmittel für einen Entwickler ist bei diesem API-Einstiegspunkt der Katalog der APIs, die direkt für die Nutzung zur Verfügung stehen. Präsentieren Sie der Außenwelt nicht jede einzelne API (es sind zu viele), sondern nur diejenigen, die für den fraglichen Entwickler von Interesse sind. Dieser Entwickler sollte sich keine Gedanken darüber machen müssen, wie und warum die API herangeschafft wurde, sondern sich darauf konzentrieren können, was er mit der API machen kann, nachdem sie ihm zur Verfügung gestellt wurde.

In einer hybriden Welt sind Domänenstrukturen notwendigerweise komplex. Öffentliche Marktplätze, private API-Kataloge, Partnerportale und vieles mehr sind Teil des API-Geflechts. Eine klar definierte Communitystruktur ist demzufolge wichtiger als je zuvor, wobei es einen direkten Zusammenhang zwischen Community-Designentscheidungen und API-Designentscheidungen gibt.

Abbildung 4-1 veranschaulicht, wie ein Entwickler APIs für den sicheren Zugriff auf beliebige Bereiche einer hybriden Umgebung verwenden kann. Während der Entwicklung werden über den API-Marktplatz Informationen zu den APIs bereitgestellt, die für die Community dieses Entwicklers verfügbar sind. Zur Laufzeit sichern die Cloud-Gateways die Kommunikation zwischen der API-Nutzerumgebung und beliebigen API-Endpunkten, und zwar unabhängig vom Standort.

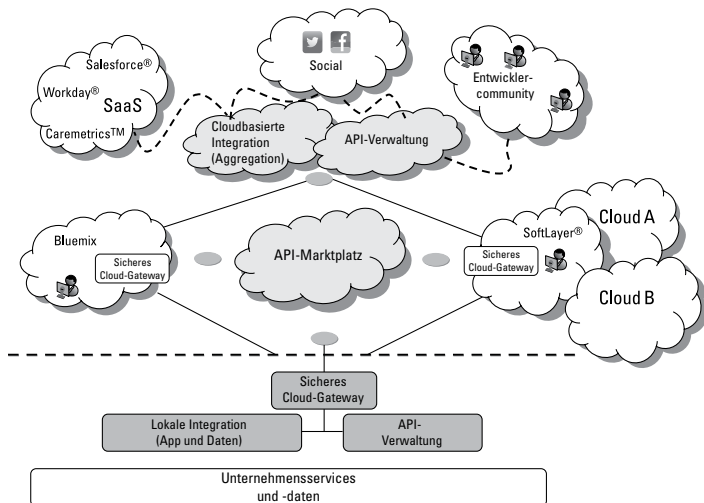


Abbildung 4-1: APIs als Verkehrssprache für eine hybride Umgebung

Hier ein paar Empfehlungen für eine API-Strategie im Kontext hybrider Umgebungen:

- ✓ **Ziel:** Das gewünschte Ergebnis ist die Unterstützung von Entwicklern und IT-Betrieb. Die Auswahl der zur Verfügung gestellten APIs muss mit großer Sorgfalt erfolgen und jegliche Kommunikation in einem offenen Netzwerk muss adäquat gesichert und kontrolliert werden.

Sicherheitsbedenken sind der am häufigsten genannte Grund für die Nicht-Einführung einer hybriden Cloudumgebung. Auf Platz 2 der genannten Gründe folgt die Angst davor, die Kontrolle über den Betrieb der Umgebung zu verlieren.

- ✓ **Zielgruppe:** Die Zielgruppe kann eine beliebige Mischung aus internen und externen Entwicklern sein, die zu dem hybriden Ökosystem gehören. Der Aufbau geeigneter Communitystrukturen ist daher außerordentlich wichtig. Entscheidungen auf der Ebene einzelner Personen zu treffen ist hierbei keine hilfreiche Vorgehensweise. Was Sie



benötigen, ist eine Struktur, die es Ihnen ermöglicht, API-Freigabeentscheidungen auf Community-Ebene zu treffen und durchzusetzen, sodass alle Entwickler in einer bestimmten Community gleich behandelt werden.

- ✓ **Bereitzustellende APIs:** Die APIs, die Sie benötigen, um die Zielgruppe zu erreichen, hängen von der Communitystruktur der Zielgruppe ab. Für externe Zielgruppen benötigen Sie höchstwahrscheinlich eine große Menge vordefinierter Unternehmens-APIs; für interne Zielgruppen benötigen Sie hingegen diverse angepasste APIs, um die schnelle Entwicklung innovativer Apps zu unterstützen. Es gibt keine verbindlichen Regeln für alle hybriden Ökosysteme, jeder Fall weist seine spezifischen Besonderheiten auf.
- ✓ **API-Nutzungsbedingungen:** In einer hybriden Welt können die Bedingungen für die Nutzung von APIs kompliziert sein, sowohl aus Business- als auch aus IT-Perspektive. Dies gilt unabhängig davon, ob die APIs verwaltet oder nicht verwaltet sind (siehe Kapitel 2); die Implementierungsmechanismen für die beiden Typen sind normalerweise jedoch sehr verschieden:
 - *Bei verwalteten APIs* wenden Sie Business- und IT-Richtlinien auf die übliche Weise auf die API-Plattform an – einer der Vorzüge der Verwendung von APIs als einheitliches Nutzungsmodell. Stellen Sie jedoch sicher, dass Ihre API-Plattform communityorientiert ist, damit Sie Entscheidungen auf der geeigneten Community-Ebene treffen können.
 - *Bei nicht verwalteten APIs* steht als einziges Mittel zur Durchsetzung von Nutzungsbedingungen die Sichtbarkeit für die Community zur Verfügung und wird durch formale Vereinbarungen ergänzt, die über Seitenkanäle festgeschrieben werden. Was nicht verwaltete APIs betrifft, verfügt IT Operations über kein integriertes Verfahren, um Sicherheitsmechanismen und Mechanismen zur Steuerung des Datenverkehrs auf API-Ebene durchzusetzen. Nicht verwaltete APIs müssen daher über sichere Tunnel aufgerufen werden, die auf Netzwerkebene hergestellt werden.
- ✓ **API-Implementierung:** Das Zusammenstellen von Daten und Funktionen für die API-Implementierung variiert in Abhängigkeit davon, ob es sich um vorab geplante Unternehmens-APIs oder angepasste, bedarfsorientierte APIs handelt: Wie im Abschnitt „Innovationsfreiheit“ weiter oben in diesem Kapitel beschrieben wurde, besteht der wesentliche Unterschied in der Fokussierung auf Zuverlässigkeit und Laufzeitkosten auf der einen Seite und

der Konzentration auf Zeitaufwand und Entwicklungskosten auf der anderen Seite.

- ✓ **Nutzung fremder APIs:** Die Entscheidung, welche APIs von Drittparteien genutzt werden sollen, ist bei diesem Einstiegspunkt schwieriger als bei jedem anderen Einstiegspunkt, da die Anzahl und Vielfalt der verfügbaren APIs mit der Zeit deutlich größer wird.



Mein Rat lautet daher, mit der API-Nutzung im kleinen Maßstab zu beginnen. Wählen Sie eine kleine Zahl wichtiger APIs aus, die Sie nutzen möchten. Ob Social-APIs, Analyse-APIs, mobile Back-End-APIs oder etwas ganz anderes, hängt dabei von Ihren unmittelbaren Business-Anforderungen ab. Darüber hinaus haben Sie die Möglichkeit, Drittanbieter-APIs in eigenen, einfacheren oder besser überwachten Versionen zusammenzustellen. Diese Möglichkeit sollten Sie im Rahmen des Entscheidungsprozesses nicht vergessen.

Hybride Umgebungen sind von Natur aus komplex. Die Verwendung von APIs als Verkehrssprache kann dazu beitragen, dass diese Komplexität aus Entwicklerperspektive etwas besser handhabbar wird. Aus IT Operations-Sicht kann die Weiterentwicklung hybrider Cloudplattformen dazu beitragen, dass diese Komplexität besser kontrollierbar wird.

Programmieren Sie Ihre Welt

Der letzte API-Einstiegspunkt konzentriert sich auf die Welt der Geräte und Maschinen. Er stützt sich auf zwei Überzeugungen: Erstens: Bei der verbrauchergesteuerten mobilen Revolution geht es nicht nur um Telefone. Und zweitens: Fertigung und Logistik treiben die Entwicklung eines intelligenten, unternehmensbezogenen Internet der Dinge an.

Ob Gesundheitswesen, Versorgungsunternehmen, Städte oder Fertigungsindustrie, wohin Sie auch blicken, lässt sich erkennen, dass ein intelligenterer Ansatz nötig ist, um Menschen, Software und Maschinen miteinander zu verbinden. Zugegeben, die Vorstellung, dass sich alles programmieren und zu einer einzigen, intelligenten Erfahrung zusammenfügen lässt, ist relativ neu und längst nicht ausgereift. Aber täglich lassen sich Zeichen des Fortschritts in Richtung dieses Ziels finden, beispielsweise intelligente Autos, interaktive Einkaufsergebnisse und umweltfreundliche Gebäude.



Ein wichtiger Unterschied trennt diesen Einstiegspunkt von allen anderen: Ein Gerät lässt sich normalerweise nicht mehr ändern, sobald es in seiner physischen Form existiert. Das bedeutet

wiederum, dass Geräte-APIs bleiben, was sie sind. Der Fokus verschiebt sich daher wie folgt auf die Bereitstellung einer ausgestalteten Benutzererfahrung mit optimierter Ausführung:

- ✓ Erweiterung der Software in die Welt des Gegenständlichen.
- ✓ Steuerung aller Komponenten, egal ob Software oder Gerät, über programmierbare APIs.
- ✓ Nutzung von gerätegestütztem Feedback und Einblicken, um das Verhalten des gesamten Systems und nicht nur einzelner Komponenten zu optimieren.
- ✓ Wo es zweckmäßig ist, sollten Sie eigene, allgemeine Steuerungsfunktionen in Form Ihrer eigenen APIs monetarisieren (Informationen hierzu finden Sie im Abschnitt „Daten monetarisieren“ weiter oben in diesem Kapitel).



Bei dem Einstiegspunkt „Programmieren Sie Ihre Welt“ geht es in der Regel sehr viel häufiger um die API-Nutzung als um die Bereitstellung neuer APIs.

Hier ein paar Empfehlungen für eine API-Strategie, wenn der Fokus auf der Programmierung der realen Welt liegt:

- ✓ **Ziel:** Das gewünschte Ergebnis ist eine vollständig programmierbare Umgebung. Suchen Sie nach einer programmierbaren physischen Infrastruktur und entwickeln Sie eigene Software so, dass sie auch über APIs gesteuert werden kann.
- ✓ **Zielgruppe:** Die Zielgruppe ist größtenteils intern. Ihre eigenen Entwickler sind diejenigen, die intelligente Systeme auf der Basis von Geräte- und Software-APIs entwickeln. Wenn Sie selbst ein Hersteller von Geräten sind, müssen Sie sicherstellen, dass Ihre Geräte über APIs verfügen, die gemäß den Anforderungen Ihres Geschäftsmodells für die Nutzung durch andere vorgesehen sind.
- ✓ **Bereitzustellende APIs:** Die APIs, die Sie benötigen, um die gewünschte Zielgruppe anzusprechen, hängen im Wesentlichen von den Schnittstellen ab, mit denen Ihre physischen Geräte und Maschinen ausgestattet wurden. Ihre Auswahlmöglichkeiten sind hier ziemlich begrenzt, da die Mehrzahl der APIs, die Sie verwenden können, vordefiniert ist.
- ✓ **API-Nutzungsbedingungen:** Wenn Sie über Netzwerkzugriff auf ein Gerät verfügen, können Sie in der Regel auch mit dem Gerät kommunizieren, da die zugehörigen APIs per Definition nicht verwaltete APIs sind. Die Nutzungsbedingungen sind bei diesem Einstiegspunkt nicht so wichtig wie bei den anderen.



Viel wichtiger ist es, über eine gut sortierte Bibliothek der verfügbaren Geräte-APIs zu verfügen.

In einigen Fällen kann es notwendig sein, den Zugriff auf die APIs auf Geräteebene zu kontrollieren. Dies ist möglich, indem Sie vor der Geräte-API eine Schicht verwalteter Proxy-APIs mit integrierten Sicherheitsmechanismen einfügen.

- ✓ **API-Implementierung:** Bei Geräte-APIs müssen Sie sich über die Methoden, die zum Zusammenstellen der Daten und Funktionen für die API-Implementierung verwendet werden, keine Gedanken machen (natürlich vorausgesetzt, Sie sind *nicht* der Hersteller von physischen Geräten).
- ✓ **Nutzung fremder APIs:** Bei diesem API-Einstiegspunkt spielt die Frage, welche Drittanbieter-APIs genutzt werden sollen, eine sehr wichtige Rolle. Es ist von entscheidender Bedeutung, mit Drittanbietern von Geräten und Maschinen geeignete Vereinbarungen zu treffen. Falls Ihnen keine aktualisierte API-Dokumentation zur Verfügung steht, ist es schlichtweg nicht möglich, effektiv mit dem Gerät zu kommunizieren.



Angesichts einer Welt, in der intelligente Geräte eine immer größere Rolle spielen, werden auch die Fähigkeiten, um diese Welt zu programmieren, immer wichtiger. Da sich diese Umgebung deutlich von der klassischen reinen Softwareumgebung unterscheidet, kann ein früher Einstieg große Vorteile bieten. Wenn Sie sich auf diese Reise begeben, sind Entwickler und Business-Verantwortliche gleichermaßen gefordert, sich damit vertraut zu machen, was es heißt, reibungslose Interaktionen zwischen Geräten, Software und Menschen zu programmieren.

Kapitel 5

API- und Integrationsmiddleware

In diesem Kapitel

- ▶ Grundlegendes zu API-Middleware
- ▶ Eine Domänentopologie auswählen

Wie in früheren Kapiteln erläutert wurde, gibt es wichtige Unterschiede zwischen verwalteten und nicht verwalteten APIs. Für eine verwaltete API gibt es adäquat durchgesetzte Business- und IT-Steuerungsmechanismen, was wiederum die Frage aufwirft, wo und wie diese Mechanismen zur Laufzeit durchgesetzt werden.

Das Durchsetzen dieser Steuerungsmechanismen im Code ist normalerweise keine gute Idee. Zum einen ist dieser Ansatz bekanntermaßen unzuverlässig, da er von der Codequalität jeder einzelnen API-Implementierung abhängt. Zum anderen muss es einer Organisation auf jeden Fall möglich sein, die Steuerungsmechanismen zu ändern, ohne die API selbst ändern und neu bereitstellen zu müssen. Anders ausgedrückt: Die Steuerungsmechanismen sollten richtliniengesteuert sein, wobei unabhängig voneinander verwaltete Richtlinien das operative (Business- und IT-) Ziel definieren und die API-Laufzeitumgebung dieses Ziel durchsetzt. Beispiele für gängige Richtlinien sind der Grad an Sicherheit, der erforderlich ist, und der Umfang des Datenverkehrs, der zugelassen wird.

Eine Middlewarekomponente, die APIs hostet und API-Richtlinien durchsetzt, wird üblicherweise *API-Gateway* genannt. Vergessen Sie nicht, dass es bei API-Plattformen nicht nur um die Laufzeit geht; sie benötigen auch Funktionalität, die zur Entwicklungszeit relevant ist.

API-Middleware ist mehr als „nur ein weiterer ESB“

Haben Sie schon einmal jemanden sagen hören: „Ich habe bereits einen Enterprise Service Bus (ESB) und bin daher für APIs bestens gerüstet“? Die logische Schlussfolgerung wäre, dass API-Middleware „nur ein weiterer ESB“ ist. Tatsächlich gibt es jedoch drei signifikante Unterschiede.

- ✓ Im Idealfall werden APIs durch Konfiguration und nicht per Programmierung definiert. Die Definition einer API per Konfiguration sorgt dafür, dass der schlanke Charakter des API-Proxys erhalten bleibt, und unterstützt die schnelle Umsetzung bei neuen und geänderten APIs. Im Gegensatz dazu sind Universal-ESB-Tools ablauf- oder codeorientiert und Serviceimplementierungen sind Teil des Bereitstellungszyklus für Unternehmenssoftware.
- ✓ API-Laufzeitumgebungen müssen extrem schnell, absolut sicher, zuverlässig und hoch skalierbar sein. Im Idealfall weist das API-Gateway routerähnliche Netzwerkeigenschaften auf, sodass das Hinzufügen eines API-Proxys als Teil einer End-to-End-Interaktion nie zu Problemen im Hinblick auf Antwortzeiten oder Latenz führt. Die schnellsten und skalierbarsten API-Gateways basieren auf domänenspezifischen, schlanken Konfigurationssprachen mit statusunabhängiger Ausführung. Obwohl Universal-ESB-Laufzeitumgebungen ebenfalls schnell, zuverlässig und skalierbar sein müssen, müssen die Ausführungsengines in der Lage sein, umfassende Kompositionen und einen gewissen Grad an Statusabhängigkeit zu unterstützen. Universal-ESB-Laufzeitumgebungen können demzufolge nicht in dem Maße im Hinblick auf Durchsatz optimiert sein.
- ✓ APIs verfügen über richtliniengesteuerte Business- und IT-Steuerungsmechanismen, die von der Authentifizierung bis zur Steuerung des Datenverkehrs reichen, sowie Geschäftsbedingungen, die die Bedingungen für die Nutzung der API regeln (der API-Plan). Obwohl einige leistungsfähige ESBs Richtlinien zur Verwaltung des Datenverkehrs einschließen, verfügen nur wenige über die Sicherheitsrichtlinienfunktionalität und Zertifizierung eines vollumfänglichen API-Gateways. Zudem verfügt kein ESB über die separaten Business-Steuerungsmechanismen, die in einem API-Plan enthalten sind.
- ✓ APIs müssen App-Entwicklern in Form einer Self-Service-Lösung zur Verfügung gestellt werden. Jede Form von

Genehmigungsprozess, der der Bereitstellung nachgelagert ist, verlangsamt die Annahme und führt letztlich zu entsprechenden organisatorischen Kosten in beträchtlicher Höhe.

Der bevorzugte Freigabemechanismus, der sich als äußerst wirkungsvoll erwiesen hat, ist die Veröffentlichung von APIs in Entwicklerportalumgebungen. Insbesondere bei der internen API-Verwendung sollte die Freigabe vorzugsweise auf Communitybasis verwaltet werden. Hierdurch kann sichergestellt werden, dass bestimmte Entwickler nur die APIs sehen, die ihre Community verwenden soll.

Universal-ESBs bieten keines dieser Features und selbst Serviceverwaltungslösungen zielen mehr auf Steuerungsmechanismen für Serviceentwicklung und -betrieb als auf die Optimierung des Freigabeprozesses ab.

- ✓ Zu guter Letzt brauchen API-Verantwortliche Geschäftsstatistiken, die Auskunft darüber geben, wer ihre APIs in welchem Umfang verwendet. Diese Statistiken messen den Erfolg anhand der Geschäftsziele für das API-Portfolio, anstatt sich auf IT-Fragestellungen zu konzentrieren, wie es die Dashboards zur Betriebsüberwachung tun, die üblicherweise in ESB-Plattformen enthalten sind.

Der ein oder andere wird vielleicht einwenden, dass nur ein einziger Bus notwendig ist, der umfunktioniert werden kann, um klassische Integrationsanforderungen, serviceorientierte Architekturen (SOA) und auch die API-Verwaltung zu unterstützen. Wenn Sie jedoch keine spezialisierten Umgebungen für API-Entwickler, API-Besitzer und API-Nutzer bereitstellen, laufen Sie Gefahr, eine Middleware für den kleinsten gemeinsamen Nenner zu konstruieren. Und falls Sie nur API-Verwaltungsfunktionen oder nur Universalfunktionen für die Integration benötigen, stellt sich die Frage, warum Sie für beides bezahlen sollten.



Moderne Unternehmen benötigen API-Verwaltungsfunktionalität sowie Funktionen für die Unternehmensintegration, dies jedoch in Form von zwei getrennten Plattformen, die sich an unterschiedliche Zielgruppen richten. Bedenken Sie jedoch, dass Sie, falls Sie bereits SOA nutzen, über eine Auswahl an Assets verfügen, die Sie in kürzester Zeit ermitteln, zu (Proxy-)APIs zusammenstellen und über eine API-Verwaltungsplattform offenlegen können.

Die (reproduzierbare) Integrationstopologie

Traditionell neigen wir dazu, uns die verschiedenen Arten von Gateways – ob es nun Web-, Sicherheits-, Messaging- oder API-Gateways sind – als eigenständige „Kästen“ in einer Bereitstellungstopologie vorzustellen. In einer hybriden Welt aus Geräten, Menschen und

Software löst sich diese Vorstellung von separaten Gatewaykästen sehr schnell in Luft auf. Vor dem Hintergrund, dass Umfang und Vielzahl der Interaktionen in erheblichem Maße ansteigen, ist es aus betrieblicher Sicht von großem Vorteil, über eine einheitliche Gatewaystrategie mit einer einzigen Kommandozentrale für den gesamten Datenverkehr zu verfügen, der interne oder externe Grenzen überquert.

Es ist bewährte Praxis, dass Datenverkehr, der eine Domänengrenze überschreitet, stets eine API durchläuft. Dieser Integrationsansatz sorgt für einen adäquaten Grad an Transparenz und Kontrolle. Darüber hinaus bietet er maximale Flexibilität im Hinblick auf den Ausgleich von Unternehmens- und angepassten Implementierungen, und zwar auf eine Art und Weise, die für den API-Nutzer nicht sichtbar wird.

Die Entscheidung für eine grob- oder eine feinkörnige Domänentopologie liegt im Ermessen des einzelnen Unternehmens. Übliche Domänengrenzen werden durch folgende Faktoren beeinflusst:

- ✓ Organisation (beispielsweise Entwickler mobiler Apps gegenüber Back-End-Team)
- ✓ Eigentümerschaft (beispielsweise Geschäftsfeldgrenzen)
- ✓ Sicherheit (beispielsweise Demilitarized Zone [DMZ] gegenüber internem Netzwerk)
- ✓ Servicequalität (beispielsweise Mainframe gegenüber dezentralen Systemen)
- ✓ Netzwerk (beispielsweise öffentlich oder Partner)

Es ist wichtig, dass das vorgeschlagene Modell für die Integrationstopologie sowohl reproduzierbar als auch zusammensetzbar ist. Jede Domäne in einem Netzwerk sollte die gleiche Topologiestruktur aufweisen, wobei domänenübergreifende Interaktionen zur Laufzeit stets ein (API-) Gateway durchlaufen, wie in Abbildung 5-1 dargestellt wird. In der Abbildung weist jede einzelne Domäne die gleiche Struktur aus konzentrischen Kreisen mit unterschiedlichen Integrationsfunktionen auf. Die Standardintegrationstopologie jeder Domäne kann domänenübergreifend zu einer System-aus-Systemen-Topologie zusammengesetzt werden, indem das Schema einfach reproduziert wird. Diese Struktur greift den System-aus-Systemen-Charakter hybrider Integrationsumgebungen präzise auf und passt gut zu jedem der API-Einstiegspunkte, die in Kapitel 4 definiert wurden.

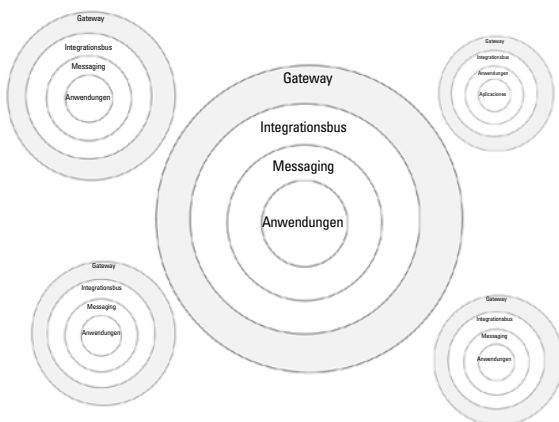


Abbildung 5-1: Die Integrationstopologie ist ein reproduzierbares Schema.

Der ein oder andere wird vielleicht einwenden, dass bei einem Schema, bei dem domänenübergreifende Interaktionen immer eine API durchlaufen, Latenz oder Antwortzeiten ein Problem darstellen können. Intelligente API-Gateways lassen Latenzzeiten und Antwortzeiten nicht zu einem Problem werden, wie im Abschnitt „API-Middleware ist mehr als nur ein weiterer ESB“ weiter oben in diesem Kapitel ausgeführt wurde. Eine Analogie kann dies veranschaulichen. Wie oft denken Sie über die Anzahl der Router zwischen einem Browser und einem Web-Server nach? So gut wie nie! Sie gehen einfach davon aus, dass die Routerinfrastruktur schnell und skalierbar genug ist, dass die Anzahl der Router kein Problem darstellt. Intelligente API-Gateways sind in dieser Hinsicht wie Router. Das Hinzufügen eines dieser Gateways auf dem Weg vom API-Nutzer zum API-Anbieter macht keinen nennenswerten Unterschied.

Verwenden Sie Universal-ESBs als Integrationsbus und stellen Sie Daten und Funktionen zu wohlgeformten Services zusammen. Definieren Sie dann (Proxy-) APIs, um zu steuern, wie diese Services im Ökosystem jenseits der Grenzen Ihrer eigenen Domäne zu Produkten gemacht werden. Leiten Sie abschließend den gesamten API-Datenverkehr durch ein hoch skalierbares, zuverlässiges und sicheres API-Gateway. Auf diese Weise wird jeder Teil der End-to-End-Integrationsplattform in dem Bereich eingesetzt, in dem seine größten Stärken liegen (siehe Abbildung 5-1).

Weitere Details zu den sich ergänzenden Funktionen einer End-to-End-Integrationstopologie finden Sie im IBM Redbook *Integration Throughout and Beyond the Enterprise* unter www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg248188.html?Open.



API und Service Economy: Referenzmodell

In den 2000er-Jahren wurde von IBM ein SOA-Referenzmodell zur Skizzierung der Middlewarefunktionen definiert, die für eine effektive Erstellung, Bereitstellung und Verwaltung von Services erforderlich sind. Diese Funktionen sind in der API Economy weiterhin von Bedeutung, da Services die Grundlage bilden, auf der API definiert werden; allerdings sind sie nicht mehr ausreichend.

Das Referenzmodell für diese erweiterte Middlewareplattform ist in Abbildung 5-2 definiert. Es bietet eine praktische Richtlinie für die Zusammenstellung an Middlewarefunktionen, die in Ihren Integrations- und API-Plattformen vorhanden sein sollten.

Die integrationsorientierten Elemente des ursprünglichen SOA-Referenzmodells von IBM werden in komprimierter Form als untere Schicht des Referenzmodells für die API und Service Economy wiedergegeben. Das neue Referenzmodell weist darüber hinaus Funktionen für die Definition und Verwaltung von APIs, für Entwicklerportale und Marktplätze sowie Funktionen auf, die die Nutzung von APIs beschleunigen.



Es ist wichtig, Entscheidungen zu den einzelnen Teilen des Referenzmodells nicht isoliert zu treffen. Verwenden Sie stattdessen eine integrierte Middlewarestrategie, um aus Assets Vorteile für das Business zu machen.

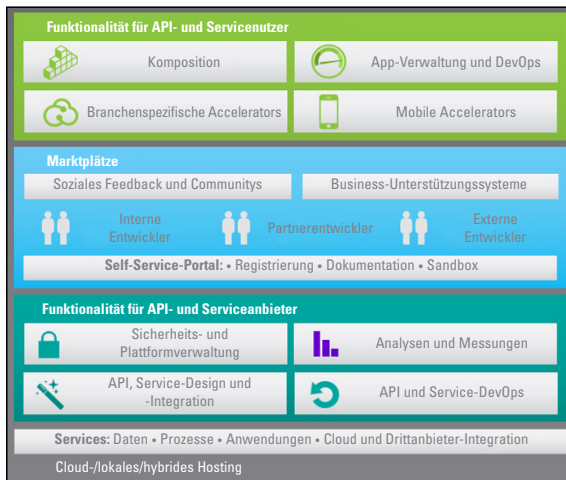


Abbildung 5-2: Das Referenzmodell für die API und Service Economy.

Kapitel 6

Zehn Dinge, die Sie über APIs wissen sollten

In diesem Kapitel

- ▶ APIs sind Produkte
- ▶ Design ohne Ende
- ▶ Jede API hat einen Besitzer

Teil des Weges hin zu APIs ist eine Denkweise, die ein Unternehmen in einer effektiven Art und Weise über APIs nachdenken lässt. In diesem Kapitel möchte ich einige Dinge vorstellen, die ich selbst auf meinem Weg durch die Welt der APIs gelernt habe.

Omnichannel-Erfahrungen fördern die Nachfrage nach APIs

Das grundlegende Konzept von APIs ist nicht neu. Was heute den Unterschied ausmacht, ist die Tatsache, dass moderne Benutzer (Verbraucher und Unternehmen) eine Omnichannel-Erfahrung erwarten, die sowohl soziale als auch personenbezogene Aspekte in sich vereint. Um wirklich personenbezogen zu sein, muss diese Benutzererfahrung zumindest bis zu einem gewissen Grad selbst instrumentiert sein. Es ist nicht mehr möglich, dass ein Unternehmen einen einheitlichen, für alle passenden Channelprozess definiert.

Das Stichwort *Selbstinstrumentierung* weist unweigerlich in die Richtung von Mikro-Apps, die wiederum die Notwendigkeit zweckorientierter APIs aufzeigen. Eine Omnichannel-Erfahrung impliziert ein Ökosystem, das Menschen, Software und Geräte einschließt, was erneut die Notwendigkeit zweckorientierter APIs verdeutlicht.

APIs sind Business-Produkte

Die Vorstellung von APIs als Business-Produkte macht es leichter, den Unterschied zwischen einem API-zentrierten Ansatz und einem klassischen Softwarebereitstellungsansatz zu erkennen. Bei einem Produkt müssen verschiedene Schlüsselfragen gestellt werden:

- ✓ Wer ist die Zielgruppe?
- ✓ Was möchte sie kaufen?
- ✓ Wie lauten die Bedingungen, unter denen ich zum Verkauf bereit bin?

Die Begriffe *kaufen* und *Verkauf* werden hier absichtlich verwendet, auch wenn die wirtschaftlichen Modelle, die APIs zugrunde liegen, stark variieren. Egal, ob der „Preis“ Geld oder Einfluss ist, ob es sich um ein Modell handelt, bei dem der Nutzer oder der Anbieter bezahlt, der produkthafte Charakter der API bleibt bestehen.

Business-Design als Herausforderung, die alle betrifft

APIs sind nicht länger ein reines IT-Thema. APIs sollten Bestandteil Ihres umfassenden Business-Designs sein.

Nehmen Sie als Beispiel eine Reisende, die ihre gesammelten Erfahrungen über soziale Kanäle mit anderen teilt. Eines Tages twittert sie über ein negatives Erlebnis mit Fluggesellschaft A. Zehn Minuten später erhält sie eine E-Mail von dieser Fluggesellschaft mit folgendem Inhalt: „Wir möchten uns für diesen Vorfall entschuldigen und Ihnen Folgendes anbieten.“ Eine Woche später macht dieselbe Frau sehr positive Erfahrungen mit Fluggesellschaft B und schreibt wie üblich einen Tweet darüber. Fünf Minuten später veröffentlicht die Fluggesellschaft einen ReTweet mit folgender Ergänzung: „Wir freuen uns, dass Sie zufrieden waren und hoffen, dass wir Sie bald wieder begrüßen dürfen.“

Beide Fluggesellschaften haben bereits im Voraus darüber nachgedacht, wie soziale Kanäle mithilfe von APIs in die Betriebsmodelle ihres Business eingebunden werden können.

Erkenntnisse durch API-Instrumentierung

Das Prinzip „Früh testen, schnell lernen und ohne großen Aufwand erweitern“ wurde schon mehrfach angesprochen. Die beste Möglichkeit, schnell zu lernen, besteht darin, Informationen anzupapfen, die bereits im Betriebssystem Ihres Business verfügbar sind. Sie können diese Informationen sehr leicht durch die Instrumentierung von APIs und die Verwendung entsprechender Funktionen für Business-Analysen zugänglich machen, die Bestandteil jeder voll funktionsfähigen API-Middlewareplattform sein sollten.

Nicht alle APIs sind REST-APIs

Ein weit verbreiteter Mythos besagt: „SOAP ist Vergangenheit; heutzutage sind APIs immer REST-APIs.“ Obwohl die meisten modernen APIs tatsächlich auf REST/JSON anstatt auf SOAP basieren, heißt dies nicht, dass es für den Formalismus von SOAP keine Verwendung mehr gibt. Es bedeutet lediglich, dass ein solcher Formalismus für die Nutzung von APIs durch Menschen (der Schwerpunkt der meisten aktuellen Erörterungen zum Thema „APIs“) nicht notwendig ist. Es gibt immer noch zahlreiche Verwendungsmöglichkeiten für SOAP im Rahmen der Kommunikation zwischen Maschinen und in Tools für die formale Komposition. Es gibt gute Gründe, warum mehr als ein Bindungsprotokoll erfunden wurde; kluge Designer wählen immer das für ihre Zwecke am besten geeignete Protokoll aus.

Jede API braucht einen Verantwortlichen im Business

Einfach ausgedrückt: kein API-Verantwortlicher im Business heißt keine Verantwortung und keine Entscheidungsbefugnis. Es ist ganz normal, dass Menschen ungern für etwas einstehen, das sie nicht vollständig kontrollieren (was sich gern in Äußerungen wie „nur dieser Teil fällt in meinen Verantwortungsbereich“ ausdrückt). Nichtsdestotrotz müssen Sie jeder einzelnen API einen Business-Verantwortlichen zuweisen. Diese Person wird damit zur Anlaufstelle bei Entscheidungen zur Produktreife und Freigabe.

APIs müssen versioniert werden

Zu behaupten, dass APIs nicht versioniert sind, ist so, als würde man behaupten, dass das Wechseln von Windeln bei Babys überflüssig ist. Störende Veränderungen lassen sich nicht vermeiden, und wenn sie geschehen, müssen Sie damit umgehen. Zu behaupten, dass eine Versionierung nicht notwendig ist, bedeutet lediglich, dass Sie es den Nutzern einer API überlassen, die Versionierung selbst herauszufinden.



Da APIs Business-Produkte sind, sollten Sie bei der Versionierung mit Sorgfalt vorgehen. Vermeiden Sie Versionschaos, indem Sie eine neue offizielle Version nur dann festschreiben, wenn ein Update nicht abwärtskompatibel ist.

APIs lassen sich mit Richtlinien leicht kontrollieren

Richtlinien sind das klassische Verfahren, wie Business- und IT Operations-Ziele festgeschrieben werden, und sorgen dadurch für Konsistenz und Integrität. Die Bedingungen, unter denen APIs genutzt und verwaltet werden, sollten in Richtlinien festgeschrieben werden, die wiederum durch API-Plattformen durchgesetzt werden.



Stellen Sie unbedingt sicher, dass Richtlinien unabhängig von der Logik, die die API selbst darstellt (beispielsweise Schnittstellen und Datenzuordnungen), geändert werden können. Dies unterstützt dynamische Änderungen am Betriebsverhalten.

APIs haben eine Schattenseite

Innovation ohne Business-Integrität ist ein schwaches Wertversprechen. Ob sich die Verletzung der Integrität in einem gebrochenen Versprechen, der Enthüllung vertraulicher Informationen oder einem unangemessenen Verhalten ausdrückt – das Ergebnis ist im Wesentlichen dasselbe. Eine schlechte Erfahrung reicht aus, damit jemand das Vertrauen in Sie verliert.



Wenn Sie eine Drittanbieter-API verwenden, müssen Sie darauf achten, dass Ihre eigene Business-Integrität keinen Schaden nimmt. Die verwendete Methode – formale Verträge mit der Festlegung von Sanktionen, Entschädigungsmechanismen oder die sachverständige Bewertung der Zuverlässigkeit und Sicherheit einer API – ist weniger entscheidend als die Tatsache, dass Sie geeignete Vorkehrungen getroffen haben. Vergessen Sie nicht, auch ethische Fragen in Ihre Überlegungen einzubeziehen.

Wie Sie die Leistungsfähigkeit von APIs für Business-Herausforderungen nutzen können

Vertrauen Sie auf die Leistungsfähigkeit von APIs! Ihr Unternehmen muss Omnichannel-Lösungen aufbauen, Innovationen schneller realisieren als die Konkurrenz und den Wandel hin zu einem mobilen Unternehmen oder die Arbeit in einer hybriden Cloudumgebung meistern. Moderne APIs bieten Ihnen die Möglichkeit, offene Ökosysteme zu verwenden und neue Geschäftsmodelle zu nutzen. Dieses praktische Buch zeigt Ihnen, wie Sie eine API-Strategie zum Vorteil Ihres Unternehmens nutzen können.

- *Betrachten Sie eine API als Produkt – als etwas, das Sie einer bestimmten Zielgruppe zugänglich machen möchten.*
- *Früh testen, schnell lernen und ohne großen Aufwand erweitern – entdecken Sie, wie Sie mit APIs experimentieren können.*
- *Nutzen Sie APIs stets als Grenze Ihrer Domäne – sorgen Sie für Kontrolle und Transparenz im Hinblick auf den ein- und ausgehenden Datenverkehr.*
- *Nutzen Sie spezialisierte API-Plattformen – sorgen Sie dafür, dass das Teilen von APIs einfach und absolut sicher ist.*



Inhalt des Buches:

- Die Anatomie einer API
- Wie werden APIs verwaltet?
- Die Beschaffenheit guter APIs
- Die API-Einstiegspunkte
- API-Middleware, die Sie benötigen
- Zehn Dinge, die Sie über APIs wissen sollten

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley's ebook EULA.