ARS Whitepaper

Die vielen Dimensionen einer API Initiative

ARS | The Art of Software Engineering

Inhalt

Abstract

Der Fortschritt der Digitalisierung in zahlreichen Ökosystemen führt zur Notwendigkeit zur Vernetzung. Daher sind API-Initiativen seit vielen Jahren ein wichtiger Teil vieler Geschäfts- und IT-Strategien.
Dieses Themengebiet fällt aufgrund des Stellenwerts für die geschäftliche Interaktion daher nicht mehr nur in den technischen Lösungsbereich einer Organisation, sondern bekommt auch zunehmend Gewicht im strategischen Produkt- und Partnermanagement.

Dieses Whitepaper beschreibt für technische und nicht-technische Zielgruppen mit strategischem oder taktischem Einfluss (z.B. C-Level Executives, Product-Manager, Enterprise Architekten, Mitarbeiter eines cross-funktionalen Product-Teams) die essenziellen Inhalte einer strategischen API-Initiative und die dafür notwendigen taktischen Handlungsfelder und Umsetzungsmaßnahmen.

Motivation

Die voranschreitende Digitalisierung unserer Gesellschaft führt zu einer zunehmend großen Menge an digitalen Services und Vernetzungsmöglichkeiten. APIs haben als neue Form der Interaktion zwischen Geschäftspartnern in den letzten 20 Jahren die klassischen menschlichen Kommunikationskanäle wie Telefon, Brief, Fax und E-Mail ersetzt.

Aber nicht nur in der Interaktion zwischen Geschäftspartnern, sondern auch für die interne Vernetzung der Systeme von cross-funktional beschaffenen Product-Teams spielen APIs eine wichtige Rolle. Verteilte Architekturen (z.B. Microservices) ermöglichen die agile Evolution der digitalen Geschäftsfähigkeiten einzelner Teams, sind aber ohne diese Vernetzungskonzepte nicht stabil umsetzbar.

Digitalisierte Geschäftsprozesse sind mit Hilfe von APIs in vielen Fällen komplett automatisierbar (Dunkelverarbeitung, Straight-Through-Processing). Neben der menschenfreien Interaktion zwischen Maschinen gibt es aber auch weiterhin die Notwendigkeit, die Expertise von Menschen mit einzubeziehen. Zu diesem Zweck ermöglichen moderne Frontends – im Sinne hocheffizienter Prothesen – die Interaktion mit technischen Kernsystemen mit Hilfe von APIs.

Vergleicht man diese drei Anwendungsfälle (Digitalisierung von Geschäftspartnerschaften, interne Vernetzung von Geschäftsfähigkeiten, Multichannel Touchpoints) miteinander, stellt man fest, dass APIs nicht nur eine technische Dimension haben, sondern auch ein Aktivator für die Vernetzung zwischen fachlich orientierten Menschen sind. Genaugenommen sind APIs damit mehr skalierendes Interaktionsmodell als Programmierkunst.

API First Strategie

API-Initiativen finden auf unterschiedlichen Flughöhen statt. Bevor man sich den taktischen Maßnahmen (Productizing, Design, Development,…) widmet, macht es Sinn, sich auf unternehmensweiter Ebene Klarheit über die strategische Bedeutung von APIs zu schaffen.

Aktuelle Forschungsergebnisse zeigen auf, dass mit Hilfe von APIs als Interaktionsmodell die beste Skalierungsfähigkeit und Entkopplung zwischen Teams und Systemen erreicht werden kann.
Zum Beispiel wird im Buch Team Topologies (Matthew Skelton und Manuel Pais; IT-Revolution 2019) das Interaktionsmodell „as a Service" und die API-basierte Selfservice Interaktion beschrieben.

APIs sind idealerweise das Ergebnis einer vorangegangenen Lernphase und Kollaboration zwischen zwei Organisationseinheiten oder Teams. Durch diese Lernphase kristallisieren sich Standards für die zukünftige Zusammenarbeit heraus, die eine lange Halbwertszeit haben und in einem API Design mit gemeinsamen Datenmodellen und Operationen münden können.

Das strategische Ziel liegt jedoch nicht nur in der Verbesserung der Interaktion zwischen Teams und Organisationen, sondern auch in der Schaffung von innovativeren digitalen Wertschöpfungsketten und Aufbau von digitalen Ökosystemen (=Vernetzung, Kollaboration).

API-as-a-Product & API Economy

APIs werden vorrangig dazu verwendet, um die Geschäftsfähigkeiten einer Organisation zu bündeln und für andere nutzbar zu machen. Dabei werden häufig Datenschätze aus schwer zugänglichen Bereichen eines Rechenzentrums gehoben und in standardisierter Form (Protokoll, Taxonomie / Datenmodell, Vereinbarungen) als Produkt angeboten und in wichtigen Ökosystemen platziert. Man spricht deswegen auch von einer „API-as-a-Product Strategie".

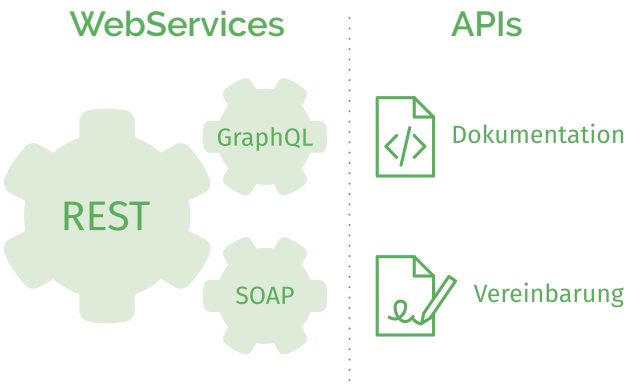
Die Art und Weise, wie eine API als Produkt behandelt wird, unterscheidet sich in der Regel je nach Art des Ökosystems allerdings sehr stark und die Anzahl der Ökosysteme ist häufig abhängig von der Größe und den technischen Anwendungsgebieten einer Organisation.

- Team-internes Ökosystem (Microservices)
- Organisationsweites/internes Ökosystem (lösungsübergreifend)
- Rechenzentrumsübergreifendes Ökosystem (Multi Cloud)
- APIs für externe Entwicklungspartner (z.B. Mobile App APIs)
- B2B Partnerökosysteme
- Edge Ökosysteme (standortbezogen, filialbezogen, Wearables)
- 3rd Party APIs

Je nach Ökosystem gelten für gleich klingende Aufgabenstellungen jeweils andere Spielregeln. Zum Beispiel weist ein API-Katalog im internen Ökosystem eher Charakteristiken eines „Developer Portals" oder „Katalogs" auf, während man im B2B-Bereich eher die Funktionalität eines „Shops" benötigt und ggf. auch Vertrags- und Monetarisierungsprozesse einen wichtigen Stellenwert haben.
Und im Gegensatz zu internen Ökosystemen, wo man Autonomie und Selfservices benötigt, spielen hingegen im B2B Partnerökosystem menschliche Beratungstätigkeiten (z.B. Partner Onboarding, Enablement und Vertragsmanagement) und zentrale Governanceaspekte eine wichtigere Rolle. D.h. was gut in einem Ökosystem funktioniert, richtet in einem anderen unter Umständen sogar Schaden an.

Aus welchen Bestandteilen setzen sich API-basierte Produkte zusammen

Technische Webservices
Eine API besteht natürlich aus einem technischen System, das eine Geschäftsfähigkeit kapselt und zum Beispiel über REST, SOAP oder andere technische Schnittstellen zugänglich macht. Aber auch asynchrone Protokolle, wie z.B. Websockets oder PubSub Protokolle wie Kafka gehören dazu. Auch diese haben logischerweise API-Charakter. Dieser Teil unserer API wird auch (Web)Service genannt.



API Design – Dokumentation und Spezifikation
Zusätzlich zu diesem technischen Teil des Systems benötigen wir in der Regel auch einen Beipackzettel für die Partner und Kollegen, die sich mit diesem technischen Dienst integrieren wollen. In dieser sogenannten API-Dokumentation bzw. dem API Design Dokument wird Folgendes beschrieben:

- Auflistung der möglichen Operationen
- Beschreibung von Taxonomien und Datenmodelle
- Erklärung der fachlichen Auswirkungen der Operationen
- Technische Spezifikation des Zugriffs (z.B. inkl. Zugriffskonzept)

Vereinbarungen und Spielregeln
Darüber hinaus gibt es auch noch weitere organisatorische Aspekte mit hoher Relevanz, die in der Dokumentation einer API geregelt werden.

- Wer sind die Ansprechpartner hinter dieser API?
- Wie kann ich mit diesen bei aufkommenden Fragen und Problemen in Kontakt treten?
- Welche Vereinbarungen und Zusicherungen seitens des API Providers gelten bei der Verwendung, wie z.B. Limits, Mengengerüste, SLAs und Verfügbarkeiten beim Zugriff, aber auch ggf. Haftungsfragen in Problemsituationen?

Standardisierung der Interaktion mit Hilfe des API Lifecycles
Wie aber kommuniziert der Anbieter einer API mit seinen Konsumenten? Hierfür hat sich ein Interaktionsmodell herauskristallisiert – der sogenannte API Lifecycle. Der API Lifecycle liefert standardisierte Antworten auf folgende Fragestellungen:

- Wie lange kann ich versichern, dass eine API-Version nutzbar ist?
- Wie veröffentliche ich Fehlerbehebungen oder zusätzliche Features, ohne die laufende Interaktion mit Konsumenten zu stören?
- Wie verhält sich ein API Provider Team im Falle von „Breaking Changes" (=Änderungen der geltenden technischen/organisatorischen Vereinbarungen)?

Also zusammengefasst sind APIs ein Interaktionsmodell, das sogenannten soziotechnischen Systemen (Wertschöpfung bestehend aus dem Zusammenspiel von Mensch und Technik) erlaubt, ihre Zusammenarbeit zu standardisieren, ohne dabei die Unabhängigkeit/Agilität beider Parteien negativ zu beinträchtigen. APIs bieten bei richtiger Anwendung die Möglichkeit, widerstandsfähige und stabile Integrationen zu ermöglichen.

Handlungsfeld API Design

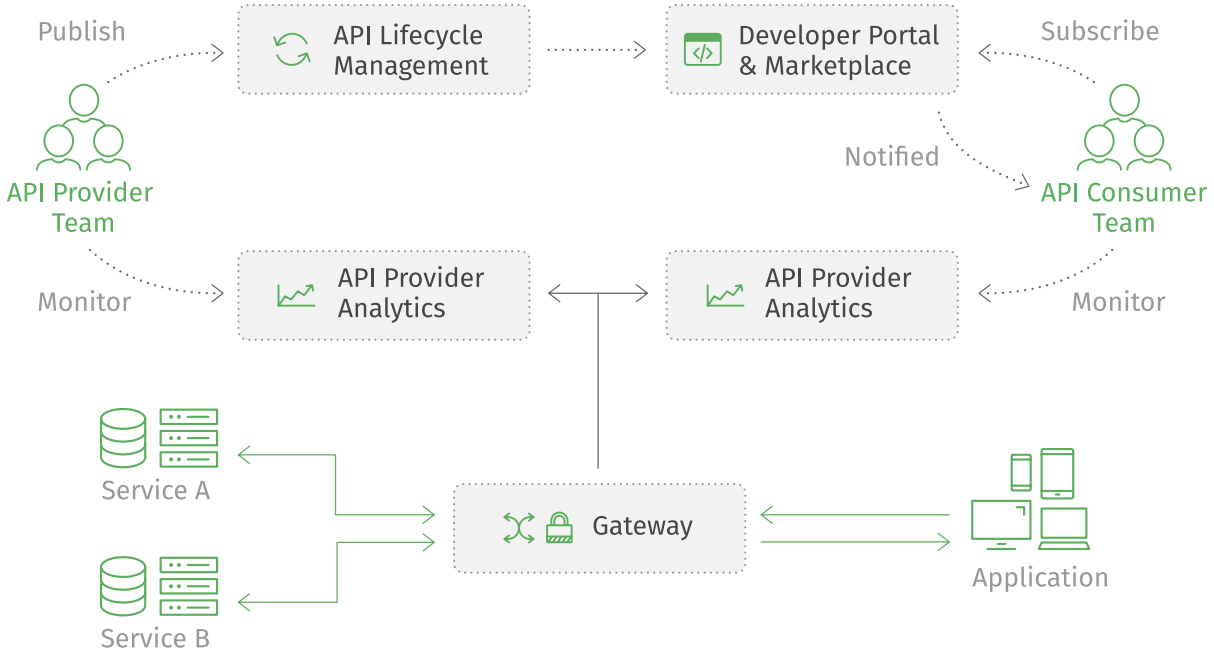
Was vermeintlich einfach klingt, ist in der Praxis die erste große Herausforderung. Ein geschickt gestaltetes und inhaltlich vollständiges API Design reduziert die Einstiegshürden und damit auch die Entwicklungskosten auf Seiten der API Konsumenten. Aber nicht nur auf Seiten der API Konsumenten bringt es Vorteile, sich hier auf Erfahrungswerte und Best-Practices zu stützen. Ein guter, fachlich versierter Schnitt der APIs, aber auch die Wahl der richtigen Design-Patterns, reduziert die Wahrscheinlichkeit von Breaking Changes und damit auch die Kommunikationsaufwände beim API-Anbieter.
Die API Spezialisten (APItekten) der ARS beraten seit 2015 zu diesem Themenkomplex sowohl Experten aus zentralen Architektur- und Technologiegremien als auch Mitarbeiter in cross-funktional organisierten Product-Teams. Die Themenschwerpunkte hierbei sind:

- Enablement zur Produktstrategie und Steuerung des Lebenszyklus von APIs
- Ausarbeitung von schlanken API Design Richtlinien für cross-funktionale Product-Teams
- DDD/Event-Storming Workshops zur Identifikation der richtigen API-Schnitte
- API Design Workshops für cross-funktionale Teams
- API Design Reviews (4-Augen-Prinzip)
- Begleitung bei der Auswahl der Werkzeuge und bei der technischen Umsetzung

Bestandteile und Dienste einer API Plattform

Diese wiederverwendbaren Bausteine sind das Fundament für eine Vielzahl neuer Geschäftsmöglichkeiten und Partnerschaften (Ökosysteme). Diese digitalen Ökosysteme werden in der Regel mit Hilfe von Marktplätzen auf einem Marktplatz realisiert. Dort können APIs veröffentlicht und angeboten und mit Ökosystem-Bewohnern interagiert werden.

Man könnte es metaphorisch auch mit dem Legospiel vergleichen. Die standardisierten digitalen APIs (Legobausteine) einer Organisation werden in entsprechende Marktplätze (Legokisten) sortiert. Unterschiedliche Ökosystem-Bewohner können sich selbstständig bedienen und mit diesen Bausteinen kreativ Lösungen entwickeln.



Bestandteile einer API Management Plattform

API Lifecycle Management
Werkzeuge zur Steuerung des Produkt-Lebenszyklus (Design, Create, Publish, Deprecate, Retire) einer API

API Gateway
Infrastruktur zur Erzwingung/Durchsetzung der vom Provider festgesetzten Regeln (z.B. Security, Rate Limits, Routing, Logging)

Developer Portal
Ein zentraler Marktplatz mit Dienstleistungen für API Consumer Teams, wie z.B. Self-Service Onboarding (API Subscription), Zugriff auf Dokumentation, Verwaltung von API Keys und Interaktion mit Communities und API Providern

Analytics
Überwachung der Nutzungsprofile der API-Produkte und Datengrundlage für Monetarisierungsvorhaben

Das API Design fungiert dabei als „Beipackzettel" oder „Bastelanleitung" und wird in der Regel in einem Marktplatz (Developer Portal, API-Shop, API-Katalog) aufgelistet. Diese Marktplätze und Ökosysteme lassen sich mit Hilfe sogenannter API Management Plattformen realisieren.
Das Ziel ist vorrangig die Bündelung komplexer/generischer Fragestellungen, um die Wertschöpfung zwischen API-Anbieter und API-Konsumenten Teams so reibungsfrei wie möglich zu gestalten. Zu den Kernaufgaben einer API Plattform zählen:

- Bündelung der Angebote für API-Anbieter und API-Konsumenten
- Standardisierung der Interaktion zwischen Anbieter und Konsument
- Steigerung der Effizienz und Geschwindigkeit der Abläufe durch Selfservices
- Absicherung der Zugriffe zur Laufzeit mit Hilfe von Gateways

Ein API-Konsument kann sich auf so einer Plattform eigenständig bewegen und wird durch unterschiedliche Dienstleistungen und Self-Services unterstützt. Eine weitere Sorte der Plattformbewohner sind die API-Anbieter (Provider). Auch für diese Zielgruppe gibt es spezifische Hilfestellungen. Folgende Tabelle bildet die typischen Fähigkeiten einer API Plattform auf die Bedürfnisse der beiden Zielgruppen ab.

Services für API Provider	Services für API Consumer
Plattform-Onboarding (per Selfservice oder per Beratungsprozess) **Plattform-Enablement und Technologieleitlinie** (Dokumentation, Videos, Links, Beratung)	
API Lifecycle Management (Create, Publish, Deprecate, Retire) **SLAs, Rate-Limits/Burst-Limits** **Security-Enforcement (Gateway)**	**Zugriff auf API-Katalog** **API Subscription als Selfservice** **Zugriff auf API-Dokumentation** **Verwaltung der API-Zugangsdaten**
Kommunikationskanäle (Blog, Forum) **Analytics und Monitoring** (Ggf. Monetarisierungsmöglichkeiten)	

API Plattform – Make or Buy

Für die Entwicklung derartiger Plattformen gibt es unterschiedliche Herangehensweisen.

Make / Build your Own
Die Lösungskomponenten werden von eigenen Plattformteams in enger Zusammenarbeit mit den beiden Zielgruppen entwickelt und betrieben.
Auf diese Weise ist es möglich, sich auf die wesentlichen Aspekte zu fokussieren und ökosystemspezifische Limitierungen und Anforderungen zentral zu lösen. Die Kosten bestehen vorrangig aus Personalkosten für Entwicklung, Betrieb und Enablement.

Buy & Customize
Viele Organisationen betrachten das Thema API Plattform als so generisch, dass sie sich für den Kauf einer Standardlösung entscheiden. Die Lösung ist von Beginn an vollständig. Die Aufwände für Installation, Anpassung und Betrieb im eigenen Rechenzentrum dürfen allerdings nicht vernachlässigt werden. D.h. auch hier entstehen nahezu die gleichen Personalkosten wie im Falle „Make", allerdings abzüglich der Entwicklungskosten.

Darüber hinaus gilt es das Risiko der „Vendoren-Falle" zu berücksichtigen, da man der Produktstrategie (inhaltliche Roadmap, Preisgestaltung, Merger & Acquisitions) des Herstellers ohne Einwirkungsmöglichkeiten ausgeliefert ist und sich entscheiden muss, wie man mit diesen Kopplungseffekten umgeht.

Rent (SaaS/Cloud vs. Managed-Services)
Im Zeitalter der Cloud gewinnt das Modell der Miete von Services an Gewicht in der strategischen IT-Budgetierung. Die Entwicklungs- und Betriebsaufwände werden vom Anbieter (Cloud oder Managed Services Provider) getragen.
Der Fokus der eigenen Mitarbeiter kann damit stärker auf Dienstleistungen für Provider/Consumer gerichtet werden. Allerdings erzeugt der Vorteil der Standardisierung durch einen Anbieter auch die gleichen Lock-In Effekte (Stichwort Vendoren-Falle) wie im Modell „Buy & Customize" beschrieben.

Handlungsfeld API Plattform

Die Möglichkeiten zum Bau einer API-Plattform sind sehr vielfältig. Die unterschiedlichen Ansätze bringen auch unterschiedliche Tradeoffs mit sich. In den vergangenen Projektsituationen sind unsere Experten mit unterschiedlichen Szenarien und Kunden in Berührung gekommen.

Wir haben einigen leidgeplagten Organisationen geholfen, sich durch einen Wechsel auf eine „Make" Strategie von der Vendoren-Schlinge zu befreien und damit stagnierenden API-Initiativen wieder neuen Schwung zu geben. Gleichzeitig haben wir auch Organisationen erlebt, die aufgrund ihrer Größe und technischen Reife nicht in der Lage sind, sich autonom eine Plattform aufzubauen und daher den „Buy" bzw. „Rent" Weg begleitet.
Unsere Experten tragen mit ihrer technischen Erfahrung und methodischen Expertise bei der Entscheidungsfindung bei:

- Klassifizierung der Ökosysteme
- Ableitung einer Plattformstrategie
- Aufbau zentraler Services und Governance-Maßnahmen
- Coaching der Plattform Mitarbeiter
- Enablement- und Onboarding-Maßnahmen für API Provider und API Consumer (Guidelines, Videos, Blueprints, Samples).
- Umsetzung prototypischer API Plattformen mit OpenSource Bausteinen zusammen mit unseren Architekten und Engineers aus den Bereichen „API-Plattform" und „Cloud"
- Einführung von Kaufprodukten und SaaS Lösungen
- Beratung zu Outsourcing / Managed Services Modellen

Die Auswahl der richtigen Technologiestrategie ist nicht banal. Ein Gießkannenprinzip über mehrere Ökosysteme hinweg kann in Einzelfällen funktionieren, führt aber häufig dazu, dass einzelne Ökosysteme zu Gunsten anderer Ökosysteme geopfert werden.
Eine genaue Differenzierung und Ausarbeitung der Tradeoffs ist daher ein wichtiger Meilenstein in jeder Plattformstrategie.

Egal ob Ihre Organisation neu in die Welt der APIs einsteigt und Sie Starthilfe beim Einstieg benötigen, oder ob eine stotternde API-Initiative neuen Schwung erhalten soll…

Unsere Mitarbeiter bringen die nötige Expertise mit, um mit unterschiedlichsten Zielgruppen (Management, Architekturbereich, Product-Teams, IT-Betrieb, IT-Security) Lösungen zu erarbeiten.
Dabei ist Expertise der ARS Mitarbeiter ausreichend breit gefächert, um auch zu angrenzenden Aufgabenstellungen zu beraten (z.B. Softwarearchitektur, Technologieberatung, Cloud Engineering, Platform Engineering, DevOps Kulturwandel). Unser Erfahrungsschatz beinhaltet daher strategische, taktische, kulturelle (= Verhalten), organisatorische, architekturelle und technische Aspekte.
So vielfältig wie unsere Aufgabenstellungen, sind auch die möglichen Formen der Zusammenarbeit. Unsere Schwerpunkte liegen derzeit auf folgenden Modellen:

- Outsourcing von (Projekt-)Gewerken & Managed Services
- Bereitstellung von eingespielten Umsetzungsteams
- Beratungs- und Trainingsdienstleistungen im Workshop-Format
- Einsatz von spezialisierten Coaches zum Enablement Ihrer Umsetzungsteams

Kontakt:

Ninette Radloff
Key Account Manager
ninette.radloff@ars.de
+49 89 3468-2025

Dr. Edgar Stoffel
Softwarearchitekt
edgar.stoffel@ars.de
+49 89 3468-2015

ARS | The Art of Software Engineering

Über ARS

ARS Computer und Consulting ist eines der führenden Unternehmen im Bereich Software Engineering. Unsere Mission: **The Art of Software Engineering**. Dies beinhaltet hochwertige Beratung und erfolgreiche Projekte zur agilen Entwicklung qualitativ exzellenter Software.
Dabei unterstützen wir unsere Kunden ganzheitlich, insbesondere in der Digitalisierung und Cloud Transformation.Unser Leistungsspektrum erstreckt sich vom Design über Architekturberatung, Entwicklung, Qualitätssicherung, Betriebskonzepte mit DevOps bis hin zu Management von APIs, Cloud-Plattformen und Künstlicher Intelligenz.

Wir begleiten Unternehmen bei der Transformation in Richtung moderne IT-Organisation, um mit neuen Arbeitsweisen, einer neuen Kultur und neuen Denkweisen die von allen Seiten geforderte Agilität vom Fachbereich über die Entwicklung und Betrieb flächendeckend auszudehnen. Die DevOps-Kultur und Arbeitsweise soll technische und organisatorische Silos und Grenzen durchlässig und die relevanten Informationen zur Wertschöpfung verfügbar machen. Die Teams erwerben sich ein breites Wissen, verbessern sich stetig durch die Perspektive auf die ganzen Prozesse und das Feedback aller Beteiligten. Wir helfen unseren Kunden, in Iterationen eine auf die eigenen Bedingungen zugeschnittene DevOps-Kultur entstehen zu lassen.

ARS Computer und Consulting GmbH
Garmischer Str. 7
80339 München
T +49 89 324 68-0
modernize@ars.de
www.ars.de

WP-API-202208-V1