



DIE API ECONOMY ERFOLGREICH MANAGEN

ALTES BASISKONZEPT - NEUE BEDEUTUNG!

Warum gibt es aktuell einen derartigen Hype um Application Programming Interfaces (API) - einer Schnittstelle für die Anwendungsentwicklung, die als Konzept so alt ist wie die Programmierung selbst? Im aktuellen Diskurs zu diesem Thema wird deutlich, dass die Abkürzung API eine komplett neue Bedeutung erlangt hat, die heute weit über der Bereitstellung einer Programmierschnittstelle für die Programmanbindung auf Quelltext-Ebene hinausgeht.

Die Grundidee ist einfach, aber auch nicht neu: Eine API ermöglicht Entwicklern, spezifische Dienste zu verwenden, ohne die eigentlichen Systeme direkt anzusprechen oder eine tiefere Systemkenntnis zu haben. Aktuelle Beispiele sind etwa APIs für Smartphones, um Hardwarekomponenten bei der Entwicklung von mobile Apps zu nutzen. So bietet YouTube Entwicklern eine API an, um nach Videos mit gewünschten Parametern suchen. Diese API liefert das Ergebnis in Form eines XML-Dokuments zurück. Somit ist ein Entwickler in der Lage, die Technologie der YouTube Plattform auf der eigenen Website oder in einer selbst erstellten App einzubinden.

Was hier entsteht, ist eine API Economy als neuartige Chance für digitale Produkte. Die Monetarisierung basiert dabei auf der transaktionalen Nutzung von API-basierenden Business Services.



Rolf Scheuch
Geschäftsführer
Opitz Consulting GmbH

Zukunftsmusik oder Chance für alle?

Ist die API Economy eher Zukunftsmusik für wenige Spezialisten? Oder geht sie in die breite Praxis? Die Antwort ist eindeutig: Keine Digitalisierung ohne API! – IT-Experten halten den Aufbau einer agilen API-basierten Systemlandschaft für entscheidend, wenn es darum geht, neue digitale Geschäftsmodelle effizient umzusetzen und schnell an den Markt zu bringen.

Bereits 2015 stellte das renommierte Harvard Business Review¹ fest, dass der Umsatz bei Unternehmen mit einem rein digitalen Geschäftsmodell zu einem großen Teil auf der Monetarisierung von APIs beruht. Hier wird zum Beispiel Expedia.de mit ca. 90 % Umsatz über APIs genannt. Das Geschäftsmodell basiert auf Netzwerkeffekten durch Partner, die Expedias Funktionalität anbieten, ihre User Experience hierdurch anreichern und sich somit differenzieren.

Ebay generiert über 60 % seines Umsatzes über Transaktionen und Auktionen, die von fremden Webseiten initiiert werden. Eine Überraschung sind die 50 % Umsatzanteil der API-Plattform von salesforce.com: Analog zum bekannten Apple Store bündeln 300 Partner ihre Angebote auf dieser Plattform. Die Lösungen basieren auf den Salesforce APIs und bieten damit selbst eine Erweiterung der Salesforce Lösung an oder verwenden einfach nur die bestehenden, robusten und grundlegenden Basisdienste von Salesforce. Ein weiteres Beispiel ist das Angebot der Reiserücktrittsversicherung der Europäischen Reiseversicherung, das uns scheinbar überall im Internet begegnet und die Angebote vieler Reiseveranstalter mit dem Mehrwert der Reiserücktrittsversicherung abrundet.

Obwohl sich Umsatzströme und Monetarisierung in diesen Beispielen sehr unterschiedlich gestalten, liegen die Hebel im Netzwerkeffekt der Angebotsplattform und in der Ermöglichung eines eigenen digitalen Angebots durch die API-Nutzung. Die API Economy ist damit im breiten Markt angekommen und steht allen, die sich entsprechend ausgerichtet haben, ready-to-go zur Verfügung.

Doch wie lässt sich nun das API-Konzept zu einer API Economy ausbauen? Und wie müssen Produktmanagement und Governance der API Economy erfolgen, um Risiken wie einen Reputationsverlust, der einen enormen wirtschaftlichen Schaden verursachen kann, zu minimieren. Dabei sollten die Fehler der zentralistischen SOA-Governance beim API-Management nicht wiederholt und Innovation, Flexibilität und Dynamik nicht durch zu viel „Red Tape“ und zentrale „Quality Gates“ zum Erliegen gebracht werden.

Der integrierte Lebenszyklus

Bevor wir ausführlicher auf den API-Lebenszyklus eingehen, betrachten wir als Ausgangspunkt zwei unterschiedliche Arten von APIs:

- Single-Purpose APIs
- Multi-Purpose APIs

Im Rahmen der API Economy stehen Multi-Purpose APIs im Fokus. Single-Purpose APIs werden weiterhin vom Entwicklungsteam der Applikation verantwortet und haben spezifische Aufgaben im Kontext einer Applikation. Somit sind Single-Purpose APIs in der Regel² nicht verallgemeinerbar und auch nicht Gegenstand einer nach Monetarisierung orientierten API Economy und deren Governance. Jedoch verwenden die Lifecycle im Normalfall ebenfalls die zentralisierten API-Runtime-Plattform und sind somit auch Gegenstand einer API Governance.

¹ <https://hbr.org/2015/01/the-strategic-value-of-apis>

² Single-Purpose APIs, die nach einem API-First-Ansatz entwickelt werden, können zu einem späteren Zeitpunkt auch für andere Apps., sowohl interne Systeme als auch für eine Monetarisierung interessant werden. In diesem Fall wird ein Produkt definiert und die Verantwortung geht an ein Produktmanagement über.

Anders als bei wasserfallartigen Vorgehensweisen mit einer ausgeprägten Vorphase der Marktanalyse, ist das Vorgehen bei APIs explorativ: Das zeigt auch der API-First-Ansatz³ bei Design und Entwicklung. Basierend auf den fachlichen Anforderungen der Applikation wird die Schnittstelle konzipiert und als Mock-up⁴ implementiert. Festigt sich die Struktur der Schnittstelle, erfolgt eine Implementierung der API mit der spezifischen Geschäftslogik. Bild 1 zeigt den Integrated API Producer- & Consumer Lifecycle im Überblick.

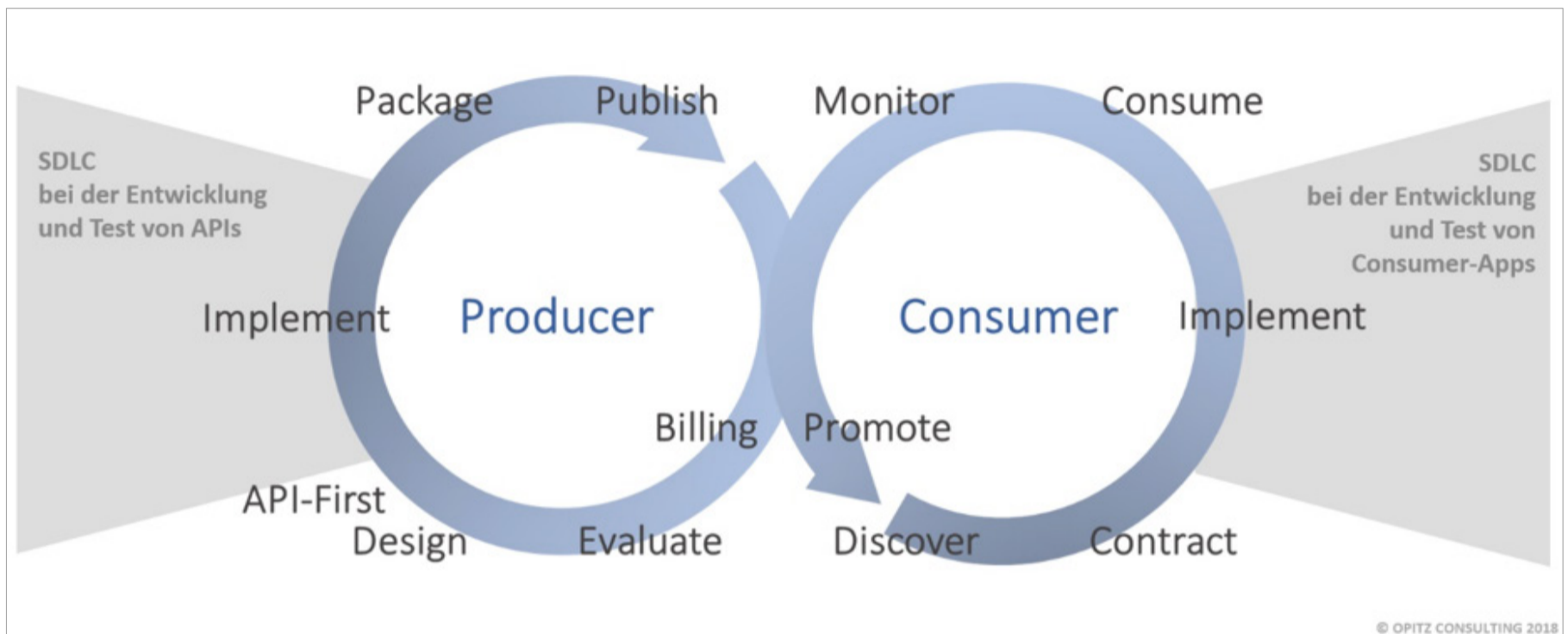


Bild 1: Der integrierte API-Lebenszyklus von Produzent und Nutzer.

Der Producer-Regelkreis

Ein API-First-Design-Ansatz lässt sich generell auch auf Multi-Purpose-APIs übertragen, um vor der Ausgestaltung der Leistungserbringung der API zunächst die einfache Nutzbarkeit und User Experience zu gewährleisten. Dies ist bereits ein MVP und kann schon mit einer ausgewählten Benutzergruppe, intern wie auch extern, getestet werden.

Erscheint der MVP von Interface-Definition und erwarteter Leistung her robust, erfolgt nun die Implementierung in einem eigenen SDLC-Ansatz. Nach erfolgreichem Test erfolgt eine Paketierung der Lösung, um diese über den API Marketplace herunterzuladen und einsetzen zu können. Die Fähigkeit, eine API als Mock-up zu nutzen, sollte auch die Nutzung durch einen Consumer vorsehen, damit er bereits frühzeitig die API einbinden und testen kann, ohne kostenpflichtige Aufrufe zu starten. Zur Paketierung gehört auch, dass die API und die benötigten Business Services zur produktiven Nutzung auf der API-Runtime-Plattform bereitstehen. Analog einer Artikellistung erfolgt nun der Eintrag der kommerziellen und Marketing-Information auf dem API Marketplace. Dies wird über ein Publish, einer expliziten internen Freigabe, abgeschlossen und sodann ist die API ein bestellbares digitales Produkt. Anschließend beginnt der Consumer-Kreislauf, wie in Bild 2 ersichtlich, wobei es selbstverständlich ein Ziel ist, digitale API-Produkte möglichst oft zu „verkaufen“. Streng genommen hat somit ein digitales Produkt pro kommerziellem Nutzer einen eigenen Consumer-Lebenszyklus.

³ <https://medium.com/adobe-io/three-principles-of-api-first-design-fa666d9f694>

⁴ <https://www.simoneabelmann.com/was-ist-eigentlich-ein-mock-up/>

In den vereinbarten Fakturierungszyklen erfolgt die Leistungsabrechnung der Verbrauchsdaten durch die Consumption der API. Auf dieser Basis erfolgt auch die Fakturierung (Billing). Der Kreislauf des Produzenten schließt sich mit der Evaluation der Verbrauchsdaten über alle Konsumenten inklusive der Leistungsdaten der API Runtime. Die Erkenntnisse der Datenanalyse haben direkten Einfluss auf die Produktentwicklung beziehungsweise auf das Produktmanagement des digitalen API-Produkts.

Der Consumer-Regelkreis

Nachdem eine API im API Marketplace freigegeben wurde, ist diese API für potenzielle Interessenten nutzbar und der API Marketplace wird eine Promotion für neue oder spezielle APIs durchführen – eine digitale Marketingmaßnahme. Ein potenzieller Anwender (Käufer) kann die API nun über eine spezielle Discovery-Funktionalität mit technischen wie auch fachlichen Kriterien oder über eine Freitext-Suche in der Dokumentation auffinden.

Zu einer aussagekräftigen und nutzbaren Dokumentation einer API gehören

- Anwendungsfälle und Best Practices als API Guide
- Referenz zu den technischen Daten (Endpoints, http-Methoden, Formate etc.)
- Installations- und Betriebshandbuch
- Vertrag mit Regelwerken der Fakturierung
- Release-Notes
- Migrationspfade bzw. -tools zu neuen Versionen

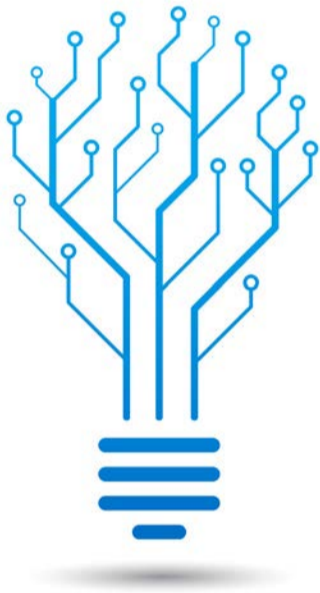
Mit der Anmeldung am API Marketplace hinterlegt der potenzielle Kunde seine kaufmännischen Daten für die Rechnungsstellung und stimmt damit den grundlegenden Vertragsbedingungen zu. Die Plattform ermöglicht hiernach ein einfaches Contracting, damit der Käufer der API-Leistung die API-Leistung in seiner Applikation konsumieren kann.

An dieser Stelle ist eine transparente, einfach zu bedienende API entscheidend für den Erfolg. Anschließend erfolgt die Nutzung der API über die eigenen Applikationen, wobei die API Runtime die Leistungsdaten über das API Gateway in ein Log File schreibt, um später einen Leistungsnachweis zu haben. Genauer betrachtet besitzt ein Consumer-Lebenszyklus auf Kundenseite auch einen eigenen Regelkreis, der eine Produktnutzung evaluiert, die Kosten der Leistung gegenrechnet und die Consumption der API verlängert oder einstellt. Das API-Management bietet Produzenten und Konsumenten auf Basis der erhobenen Daten ein Monitoring der Leistungsdaten an.

Anmerkung: Wir haben bewusst den Begriff API-Management etwas in den Hintergrund versetzt, da dieser Begriff zum einen ein zentrales API-Management impliziert und zum anderen die Tool-Hersteller diesen Begriff auf ihren angebotenen Tool Stack reduzieren.

Szenarien

Betrachten wir nun einen Lebenszyklus aus Sicht der API-Nutzung, so lassen sich, wie auch in Bild 2 dargestellt, vier Szenarien der Nutzung und somit Ausprägungen des Lebenszyklus festhalten, wobei selbstverständlich Mischformen die Regel sein werden. Trotzdem wollen wir die einzelnen Szenarien getrennt betrachten, da die Zuständigkeiten sehr unterschiedlich sind.



Die interne Nutzung als Consumer von internen APIs

Die interne Nutzung von eigenen APIs als Consumer erfordert eine komplette Ausprägung des Lebenszyklus. In diesem Fall wird eine API als digitales Produkt mehrfach von unterschiedlichen, internen Applikationen als Consumer verwendet. Der Software Development Life Cycle (SDLC) für die Applikationen, die als Consumer dienen, sind nicht notwendigerweise beim Entwicklungsteam der API (also im Produktmanagement für das digitale Produkt) angesiedelt, da eine Multi-Purpose-API in beliebigen Kontexten wiederverwendet werden kann und auch soll⁵. Jedoch obliegt dem Produktmanagement der verwendeten API die Überwachung der Nutzung sowie bei Bedarf eine interne Leistungsverrechnung.

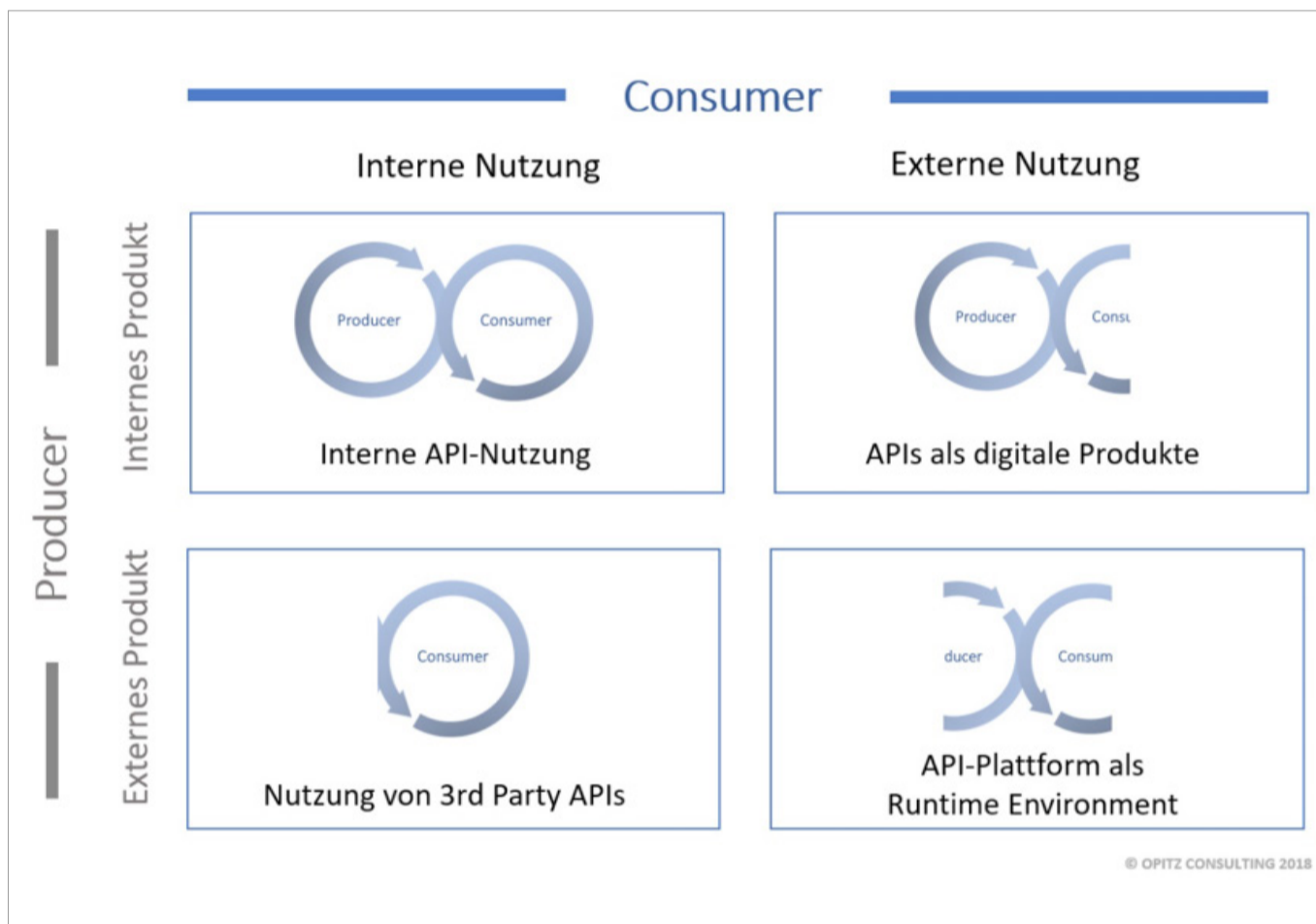


Bild 2: Nutzung von APIs und relevanter Anteil am Lebenszyklus.

Die interne Nutzung von 3rd Party APIs

Bei der Nutzung von 3rd Party APIs ist der Lebenszyklus erheblich eingeschränkt und die Verantwortung obliegt dem Owner der Applikation, die als Consumer der 3rd Party API auftritt. Bei der Einhaltung der notwendigen GRC, unter die auch die Sicherheitsregularien des Unternehmens fallen, unterstützt eine zentrale Governance. Diese Governance kann sich sogar auf eine ethische Bewertung des Einsatzes der fremden API ausdehnen, wie etwa die Nutzung von Social Media APIs. Generell ist in diesem Fall kein Produktmanagement für die fremde API nötig.

Herausfordernd wird ein Interessensausgleich über die API Governance, da eine 3rd Party API von unterschiedlichen fachlichen Applikationen eingesetzt werden kann und unterschiedliche Nutzungsarten derselben 3rd Party API bestehen können. Eine API Lineage, die Verwendungsnachweise nun übergreifend zusammenstellt, wird zu einem notwendigen Baustein. An dieser Stelle helfen zudem Standardkomponenten für ein API-Management, um ein Monitoring der 3rd-Party-Aufrufe durchzuführen und über Log-File-Analysen eine Evaluation der Nutzung zu ermöglichen.

⁵ Betrachtet man jedoch Multi-Purpose APIs, die Basisdienste wie etwa Autorisierung/Authentifikation anbieten, so kann auch die zentrale IT der Owner sein und diese Basisdienste als Shared Services anbieten.

Die externe Nutzung von internen APIs

Die externe Nutzung von internen APIs ist das klassische Szenario der API Economy mit einem eigenen digitalen API-Produkt als externem Angebot. Der Lebenszyklus beinhaltet in diesem Fall nicht den SDLC der fremden Applikationen als Consumer. Jedoch misst das API Gateway die Consumption, die Grundlage der Monetarisierung ist.

Der Betrieb einer API-Runtime-Plattform für Dritte

Das vierte Szenario ist eher ein Spezialfall. Das Unternehmen bietet nicht notwendigerweise eigene APIs an, stellt aber eine API-Runtime-Umgebung Dritten zur Verfügung, etwa als Platform-as-a-Service (PaaS). Typische Beispiele sind die Plattformen von Apple, Amazon oder Google, die neben der Runtime auch einen API Marketplace anbieten. Ist die Nachfrage groß genug, treten auch die erhofften Netzwerkeffekte ein.

In der Praxis kommt es zu einer Vermischung der unterschiedlichen Szenarien. Über die API Economy im Unternehmen sind nun allen Beteiligten die Leitlinien der API Governance bekannt. Außerdem akzeptieren alle die Notwendigkeit einer zentralisierten Plattform.

Organisation der API Economy

Im Rahmen des Managements einer API Economy kommen drei neue Aufgaben auf die Unternehmen zu (siehe Bild 2):

1. **Die übergreifende Governance einer API Economy mit einem Benefits- und Risikomanagement:** Es besteht keine Notwendigkeit eines zentralisierten Portfoliomanagements für die digitalen Produkte, da die Peripherie die Verantwortung über den wirtschaftlichen Erfolg wie auch die Innovationsfähigkeit zur Invention neuer digitaler Produkt durch APIs besitzt⁶. Die Governance einer API Economy sollte beim CDO als Stabsstelle implementiert werden, um eine unabhängige Position zur wirtschaftlichen Verantwortung der operativen Einheiten und der Betriebsverantwortung der zentralen IT einzunehmen.
2. **Das Produktmanagement der einzelnen digitalen API-Produkte:** Das Produktmanagement umfasst den gesamten integrierten API-Lebenszyklus als Producer wie auch als Consumer (Abbildung 3). Da APIs in unterschiedlichen operativen Einheiten erfunden, implementiert und weiterentwickelt werden, liegt auch die Zuständigkeit des Produktmanagements in unterschiedlichen dezentralen Einheiten.
3. **Das Management einer, meist zentralen, API-Runtime-Plattform:** Bei dieser Aufgabe geht es darum die skalierbare und elastische Plattform als Laufzeitumgebung für die APIs sicherzustellen. Insbesondere die Sicherheitsaspekte sollten durch eine zentrale Einheit überwacht und forciert werden. Diese Aufgabe obliegt in der Regel der zentralen IT und ist im Bereich der Betriebsführung angesiedelt

⁶ Sollte ein Fachbereich nicht die notwendige Innovations- und Gestaltungsfähigkeit besitzen, so sollte nicht eine zentrale Einheit einspringen, sondern der Fachbereich sollte durch Digital Awareness Initiativen befähigt werden, um selbstständig an der Peripherie die richtigen Entscheidungen einzuleiten.

Das Management der API-Runtime-Plattform ist eine typische Betriebsaufgabe der IT, die deshalb im Folgenden auch nicht weiter vertieft wird.

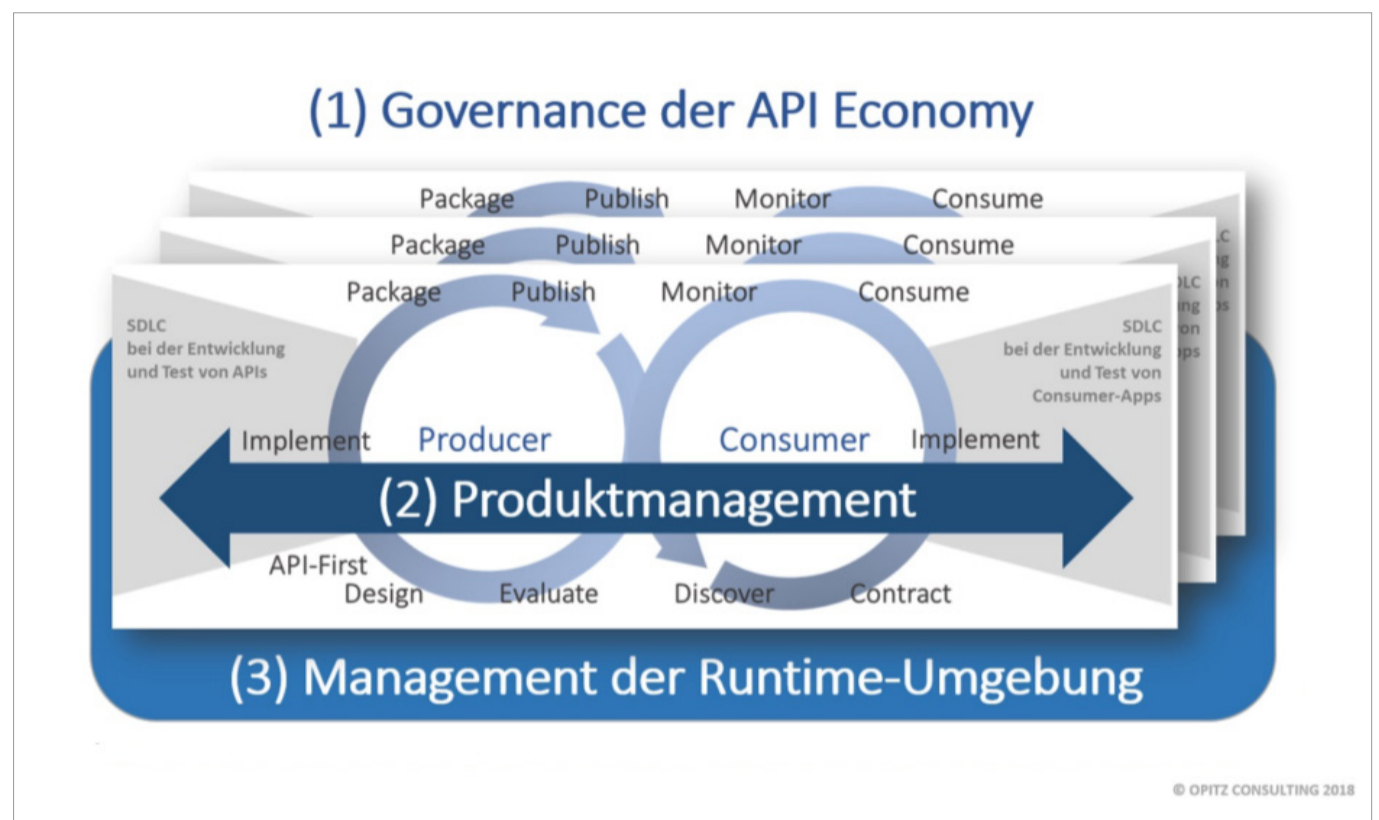


Bild 3: Die drei wesentlichen Managementaufgaben bei der API Economy.

Governance

Die benötigte Geschwindigkeit für die Einführung neuer Lösungen, die Anpassung bestehender digitaler Produkte bei einer hohen Marktdynamik wie auch die Beherrschung der bestehenden Komplexität sind über eine zentrale IT alleine nicht mehr leistbar. Wir müssen mehr Verantwortung und Selbstorganisation in den Ort der Wertschöpfung legen: In die Peripherie mit den operativen Einheiten im Unternehmen.

Die IT muss dies proaktiv aufgreifen und das entstehende dezentrale, föderale IT-Ökosystem managen und über eine angemessene Governance steuern. Diese Sicht betrifft insbesondere die digitalen Produkte der API Economy. Dies wiederum führt zu einer angepassten Sicht auf die Entwicklungs- und Einführungsmethodik. Die Einführung agiler Methoden bei innovativen beziehungsweise differenzierenden Systemen wird notwendig und die Beschleunigung der Auslieferungen neuer Release-Stände greift DevOps-Ansätze auf. Das Portfoliomanagement für APIs wird immer weniger alleine durch die IT bestimmt. Die isolierte Projektsicht weicht einem Produktmanagement unter Regie der Fachbereiche, dass je nach Lebenszyklus entsprechende Methoden und Vorgehen wählt.

Eine grundlegende Leitlinie der Vergangenheit war die Kontrolle über eine weitgehend zentralisierte Plattform. Im Mittelpunkt stand die Betrachtung der IT als Cost Center. Diese Sicht auf die Rolle der IT verändert sich aktuell, da IT-Organisationen durch die Digitalisierung unter Druck geraten. Der Grund: Die zentralistische IT kann nicht in der geforderten Geschwindigkeit liefern und die, meist im Vorjahr, vereinbarten IT-Budgets lassen die notwendige Flexibilität, Neues aufzugreifen, nicht zu.

Dieses Denkmuster passt perfekt zu unserer Sicht auf die API Economy. Dezentrale Ansätze mit einem Produktmanagement im Fachbereich und dem Schwerpunkt auf kundenzentrischen Geschäftsprozessen bzw. digitalen Produkten wird man in Zukunft immer häufiger antreffen. Die aufkommende „Schatten-IT“ setzt sich zunehmend mit dem Argument der zu sichernden Wettbewerbsfähigkeit über dieses, oft zu starre, Regelwerk der zentralen IT hinweg.

In Bezug auf die Führung und Steuerung einer eher dezentralen Wertschöpfung in einer API Economy benötigen wir eine leichtgewichtige Lösung für eine zentrale Governance, die eine ganzheitliche und an der Digitalisierungsstrategie ausgerichteten Sichtweise einnimmt. Neben Innovationsthemen bzgl. neuer APIs und produktionsrelevanter Themen einer zentralen API-Runtime-Umgebung fließen auch Überlegungen zu Synergien bei eingesetzten Technologien bzw. Architekturen ein. Jedoch ist nicht eine operative Durchführung von Vorhaben die wesentliche Aufgabe: Vielmehr stehen Koordination, Kollaboration und das eher qualitative Controlling der Aktivitäten im Mittelpunkt.

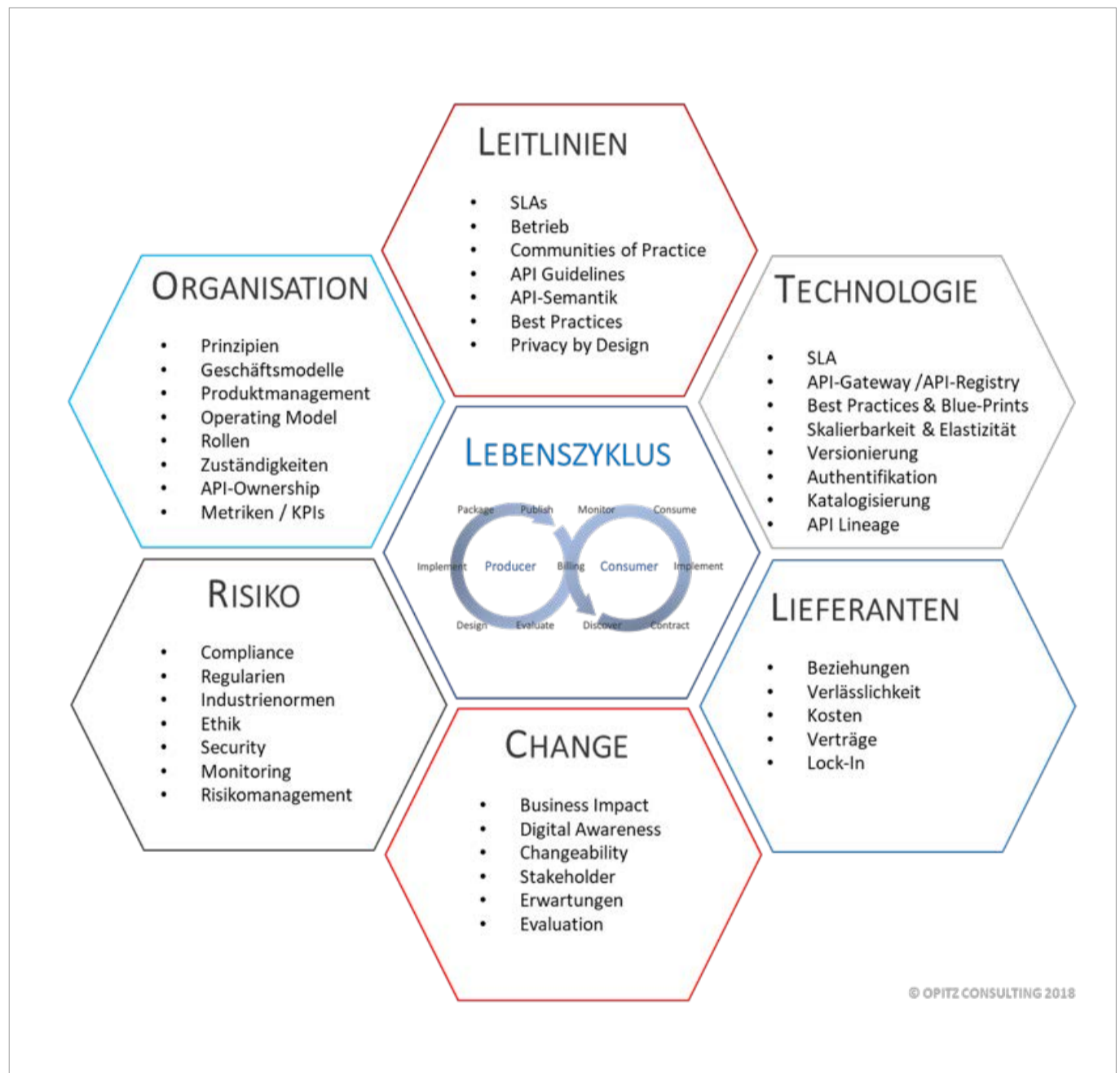


Bild 4: Ein Referenzmodell für die Governance der API Economy.

Die API Governance sollte als Stabsstelle unter dem Vorstand bzw. der Geschäftsleitung angesiedelt werden. Aber Achtung: Implementiert man eine übergreifendes Governance als Vorstandsressort, kann es zu Interessenskonflikten kommen, da das Ressort eigenes Personal aufgebaut hat und dieses somit beschäftigen muss. Sollte ein CDO etabliert sein, so wäre der CDO „accountable“⁷ für das Thema API Economy und somit der Leiter der Stabsstelle. Neben einer Stelle des Portfoliomanagers für die API Economy wäre der Stabsstelle ein kleines Team fest zuzuordnen.

⁷ <https://project-management.com/understanding-responsibility-assignment-matrix-raci-matrix/>



Weiterhin sollte ein Lenkungsausschuss implementiert werden, um Portfolioentscheidungen und das Risk- und Benefits-Management zu begleiten und abzusegnen. Somit sitzen im Lenkungsausschuss neben dem CDO auch die Leiter der betroffenen Sparten beziehungsweise der operativen Einheiten und der IT-Verantwortliche für die API-Runtime-Plattform. Ferner verwendet das Portfoliomanagement neben Innovationsvorgaben die generellen GRC-Richtlinien sowie die Digitalisierungsstrategie des Unternehmens als Input.

Die Governance einer API Economy erhält nun eine strategische und operative Facette. In Bild 4 finden sich die grundlegenden Themen der API Governance, um die Breite des Aufgabengebiets kenntlich zu machen. Zukünftig werden Verfügbarkeit einer digitalen API-Runtime Plattform, Bereitstellung notwendiger Integrationsleistungen und grundlegende nichtfunktionale Vorgaben an die API Economy zu zentralen Aufgaben der IT. Gerade der Ansatz „Privacy by Design“⁸ darf von zentraler Stelle nachhaltig forciert werden, da Reputationsverluste für das Unternehmen durch eine mangelnde API-Sicherheit erheblich sein können^{9 10}. Damit diese Prozesse nicht zu schwergewichtig werden, sollte frühzeitig an einer Automatisierung zur Beschaffung notwendiger Informationen und Zertifikate gedacht werden.

Eine API Governance hat Einfluss auf andere Initiativen der Unternehmens-IT und muss diese entsprechend unterstützen:

- Technisches Change und Configuration Management mit ITIL-Prozessen und Vorgaben zur Versionierung. Hier kommen DevOps-Ansätze zur Automatisierung zum Einsatz, die eine hohe Veränderungsrate von APIs, gerade im IT-Betrieb, ermöglichen.
- Asset Management, um einen Überblick über die bestehenden IT-Assets in den digitalen API-Produkten zu erhalten, aber auch die digitalen API-Produkte selbst als Asset zu verwalten.
- SOA Governance, um die SOA Services und die Multi-Purpose-APIs in einem gemeinsamen Portfolio transparent darzustellen und zudem über eine API Lineage die Nutzung von SOA Services in den digitalen API-Produkten aufzuzeigen.
- GRC, IT Auditing and Quality Audits mit einer transparenten Dokumentation, um belastbaren Laufzeitanalysen zu unterstützen

Produktmanagement

Das Paradigma „You build it, you run it“ bei der Entwicklung und Wartung einer API impliziert ein Umdenken hinsichtlich der Zuständigkeiten und der Aufgaben der IT in Bezug auf die API. In der Regel gibt es bei klassischen aktuellen Hand-Over- und Quality-Gate-orientierten Ansätzen (siehe auch V-Modell, PRINCE2 oder ITIL-Ansätze) bislang eine strikte Trennung von Entwicklung, Qualitätssicherung, Testmanagement und Betrieb:

Wie am Fließband übergibt ein Bereich sein Gewerk an die nächste Stelle und kommuniziert die Fertigstellung mit einer Abnahmeprozedur und oft recht aufwendigen Quality-Gates. In der Folge erlahmt die IT-Organisation an den internen Prozeduren und Korrekturschleifen. Der typische Ausweg sind meist wenige, aber dafür entsprechend große Releasewechsel – oft nur einmal pro Halbjahr. Hierdurch verlängert sich die Time-to-Market einer neuen Idee und den Fachbereichen geht wertvolle Zeit für Innovationen, neue digitale Produkte und eine verbesserte Wertschöpfung in den Geschäftsprozessen verloren.

Dies trifft auch auf den bislang eher zentralistischen Ansatz bei der API-Entwicklung zu. Somit müssen wir bei der API Economy neue Wege gehen und die Dezentralität stärken: Jede API ist ein digitales Produkt und benötigt somit ein dezentrales Produktmanagement!

⁸ <https://www.datenschutzbeauftragter-info.de/was-bedeutet-privacy-by-design-privacy-by-default-wirklich/>

⁹ <http://www.faz.net/aktuell/wirtschaft/facebook-datenskandal-doch-mehr-als-87-millionen-betroffen-15546851.html>

¹⁰ <https://www.digitaltrends.com/social-media/what-facebook-users-should-know-about-cambridge-analytica-and-privacy/>

Wir benötigen eine Transformation von einem eher klassischen Wasserfall-Modell mit definierten Quality-Gates und einem dokumentierten Hand-Over-Verfahren zu einem agilen Entwicklungsansatz, der über die DoD (Definition of Done) eigenverantwortlich die Qualität sichert.

Diese Sichtweise setzt ein Produktmanagement für das digitale API-Produkt einschließlich aller notwendigen Systeme für die Leistungserbringung voraus. Dies ist wichtig, da die API als Programming Interface alleine keinen Mehrwert darstellt, sondern erst mittels der angesprochenen Geschäftslogik, meist über Business Services im Backend implementiert wird. Da die API als digitales Produkt nun eine operative Verantwortung benötigt, sollte das Produktmanagement beim Fachbereich liegen und der Product Owner für die Weiterentwicklung der API verantwortlich sein.

Das Produktmanagement hat die Aufgabe, kontinuierlich den Wertbeitrag des Produkts über den Lebenszyklus zu sichern, an die Anforderungen anzupassen und ein Monitoring zu verantworten. Nach diesem Verständnis ist der Application Lifecycle Management (ALM) einer API mit dem Management eines Produktlebenszyklus vergleichbar. Insbesondere ist hierdurch das ALM produktzentrisch am Nutzen und der Wertschöpfung orientiert, so wie es auch ähnlich dem Produktmanagement den Mehrwert eines Produkts für den Anwender oder Käufer in den Vordergrund stellt. Dies steht im Gegensatz zu einem projektzentrischen Lifecycle Management, das den Erfolg als Folge von isolierter (Teil-)Projekte anstrebt und heute fast schon überholt ist.

Daher müssen IT und Fachbereich als Partner mit einer gemeinsamen Verantwortung ALM als kontinuierliche Produktentwicklung begreifen. Tabelle 1 stellt die wesentlichen Unterschiede der Betrachtungsweisen gegenüber und zeigt die organisatorischen Implikationen der unterschiedlichen Sichtweisen auf. Ausgehend von einer auf Effizienz ausgerichteten Sicht auf ein ALM, erfolgt ein Übergang hin zur Steuerung von Effektivität einer gemeinsamen Lösung, dem digitalen API-Produkt. Im Produktmanagement wird der Fachbereich für den wirtschaftlichen Erfolg verantwortlich (accountable) sein.



Kriterium	Projektzentriert	Produktzentriert
Kernaufgabe	Projekt in Time, Cost & Budget	Flexibilität zur Optimierung des Wertbetrags ermöglichen
Anforderungen	Pflichtenhefte aus Fachbereich	Gemeinsam sinnvolle Anforderungen erarbeiten
Entwicklung/Wartung	Getrennt vom Fachbereich und Reporting an übergeordnete Stelle	Teil der Aufgabe des Fachbereichs
QM/Testaufwand	Nur bezogen auf das Projekt und die Anforderungen	Automatisierung und am Kundennutzen ausgerichtet
Zielerreichung	Qualität und Effizienz (Kosten)	Wertbeitrag und Effektivität
Wirtschaftlichkeit der Lösung	Nicht relevant	Relevant

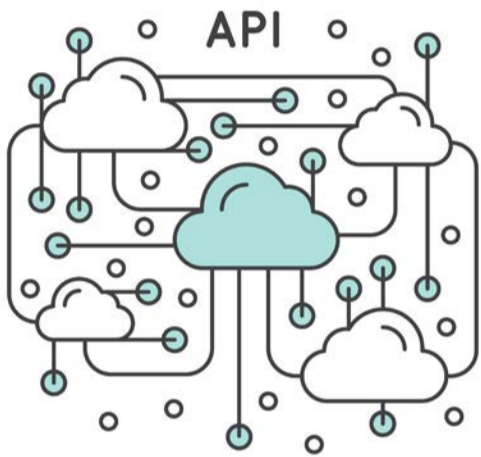
Tabelle 1: Unterschiede Projekt- und Produktmanagement.

Ein gutes Kriterium für die Notwendigkeit eines Produktmanagements ist das Vorliegen einer internen oder externen Verrechnung der Leistung einer API. Ist eine Verrechnung nicht existent, so existiert auch kein digitales Produkt und es besteht keine Notwendigkeit für ein Produktmanagement.

Ausblick

APIs sind zu einem zentralen Baustein für die Digitalisierung und den Aufbau einer dynamikrobusten Systemwelt¹¹ geworden. Der zunehmende Wunsch nach Vernetzung und Automatisierung verstärkt die API-Nutzung. Die Zeit ist reif, nicht nur APIs zu nutzen, sondern eine API Economy im Unternehmen erfolgreich aufzubauen und zu führen.

An Stelle eines weiterführenden Fazits wage ich auf der Basis unserer Erfahrungen eine Prognose zu den teilweise schon absehbaren Entwicklungen der API Economy beziehungsweise der API-Nutzung. Hierzu stellen wir folgenden Thesen auf:



1. **Die Welle der Open (Public) APIs nimmt ab, da mehr und mehr Firmen aus einer API-Nutzung monetäre Vorteile ziehen wollen:** Die Entwicklung im E-Government-Bereich wird gegenläufig sein, da die Open Data Policy der Länder und Gemeinden zu einer verbreiteten Nutzung öffentlich zugängiger Daten führt. Ansonsten werden kostenfreie Public APIs mehr und mehr „Freemium Modellen“ weichen, die nur eine Basisfunktionalität anbieten. Zusätzliche Mehrwert-Dienste werden darüber hinaus kostenpflichtig vertrieben.
2. **Die Digitalisierung und der Wunsch, neue digitale Produkte zu generieren treibt die API Economy an:** Es ist noch unklar, ob es in Zukunft eine Vielzahl an unabhängigen API-Marktplätzen als Aggregatoren und Brokers geben wird oder ob sich wegen der Netzwerkeffekte doch nur wenige, aber sehr umfangreiche, API-Marktplätze durchsetzen werden. Sicher scheint, dass viele Unternehmen über einen fachlichen oder branchenbezogenen Marktplatz versuchen werden, die Kontrolle über ein Marktsegment oder eine Nische zu erhalten und an den Mehrwerten der Teilnehmer zu verdienen.
3. **Conversational APIs werden immer wichtiger:** Mit dem Aufkommen von Chatbots braucht man APIs, die natürliche Sprache verarbeiten, um automatisierte Systeme zu erstellen. Diese Systeme basieren künftig aufgrund der eingesetzten KI-Komponenten wahrscheinlich eher auf einem Cloud-Liefer- und Servicemodell und werden somit ein Baustein der hybriden Architektur einer API Economy.
4. **APIs verstärken die Microservices- und SaaS-Nutzung:** Neben der verstärkten Nutzung von APIs für den transparenten Zugriff auf Legacy-Systeme im Zuge einer Applikationsmodernisierung werden APIs an Bedeutung gewinnen, einerseits als Verbindungstür zu Microservices, andererseits als Methode für die Applikationsintegration zur Verbindung von SaaS-Lösungen. Insbesondere wird die Integrationslandschaft um ein Netzwerk an API-Aufrufen bereichert.
5. **APIs als Backbone einer Hyperconnected World:** Mit dem Internet of Everything und den deutschen Industrie-4.0-Bestrebungen führt der Informationsaustausch von Systemen und Dingen untereinander zu einem Wachstum bei den APIs. Diese API-Welt muss kontrolliert und gesteuert werden. Insbesondere wird das Thema der API Lineage, den Verwendungsnachweisen der API-Nutzung, eine besondere Bedeutung zukommen, da Unternehmen ein Metadaten-Repository benötigen, um die Vielzahl an APIs automatisiert verwalten und deren Beziehungen analysieren zu können.

¹¹ <https://www.opitz-consulting.com/portfolio/digitalisierung/e-book-dynamikrobuste-architekturen-der-digitalisierung.html>

Rolf Scheuch