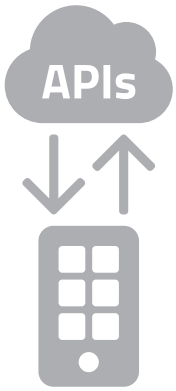
API-Management und CIAM – das neue Traumpaar für digitale Geschäftsmodelle?

Der digitale Raum bietet ideale Bedingungen für die Transformation klassischer Geschäftsmodelle, aber auch für die Entwicklung ganz neuer Ansätze. Voraussetzung dafür ist ein zuverlässiges, sicheres und komfortables Identitätsmanagement. Aus Nutzersicht noch wichtiger sind die angebotenen Dienstleistungen. Sie ganz nach eigenen Bedürfnissen zu kombinieren, sich ein eigenes digitales Ökosystem zu schaffen und dabei mit einer Identität nahtlos zwischen verschiedenen Apps, Diensten und Anbietern wechseln zu können, das verspricht die Kombination von CIAM und API-Management. Dass damit auch auf Anbieterseite enorme Vorteile und Potenziale verbunden sind, erläutern wir in diesem Whitepaper.

Schon 2006 formulierte der britische Unternehmer Clive Humby das Schlagwort „Daten sind das neue Öl". Noch vor der Markteinführung des ersten iPhones wurden damit die Claims der Digitalbranche neu abgesteckt. Seitdem ist das Datenaufkommen exponentiell gewachsen. Im weiter stark expandierenden Online-Handel, in der immer breiteren Nutzung von immer vielfältigeren Social Media Plattformen und nicht zuletzt auch im B2B-Bereich, wo die jederzeitige Verfügbarkeit und Analysierbarkeit von Unternehmensdaten den wirtschaftlichen Erfolg entscheidend mitbestimmen. Trotzdem bleibt die Frage: Wenn Daten das neue Öl sind, wo ist dann die Tankstelle? Wo wird aus dem Rohstoff ein entsprechend aufbereitetes, konsumierbares Gut? Die Antwort ist einfach: an den Schnittstellen zwischen Nutzern und Serviceanbietern, an den APIs.



Heutzutage laufen die meisten Apps nicht mehr nur auf dem lokalen Gerät, sondern sind eng gekoppelt mit APIs, die in der Cloud bereitstehen.

Ein kurzer Blick zurück

Aus Sicht der Informatik sind APIs eigentlich keine neue Idee. Letzten Endes geht es um verteilte Systeme. Ansätze gab es bereits in den 1990er Jahren mit CORBA, dies war technisch jedoch sehr komplex. In den 2000ern verlagerte sich der Fokus dann mehr auf serviceorientierte Architekturen und ihre Umsetzung mittels Java Enterprise Edition, SOAP oder DCOM. In der Regel betrafen diese API-Ansätze noch serverseitige Geschäftsprozesse. Erste Web-APIs sind als Teil von AJAX-Architekturen im Rahmen von dynamischen Webseiten entstanden und haben dann mit Mobile Apps gegen Ende des Jahrzehnts massiv an Bedeutung gewonnen. Mit dem Wechsel vom Desktopsystem zum Mobile Computing ging eine rasante Entwicklung von Web-Ökosystemen einher. Heutzutage laufen die meisten Apps nicht mehr nur auf dem lokalen Gerät, sondern sind eng gekoppelt mit APIs, die in der Cloud bereitstehen. Tatsächlich generieren viele Apps ihren Nutzen daraus, dass sie Dienste unterschiedlicher Anbieter

unter einer Oberfläche verbinden und daraus einen Mehrwert schaffen. Etwa wenn eine Touristik-App die Hotelbuchung mit Kartendiensten und anderen Serviceleistungen kombiniert. Je nach Anreiseform erhalten Kunden dann z.B. eine individuelle Anfahrtsroute oder geeignete Mietwagenangebote.

Zugriff zu regeln und Ordnung in das Ganze zu bringen. Genau das leistet ein API-Management.

Von der Ordnung zum Ökosystem

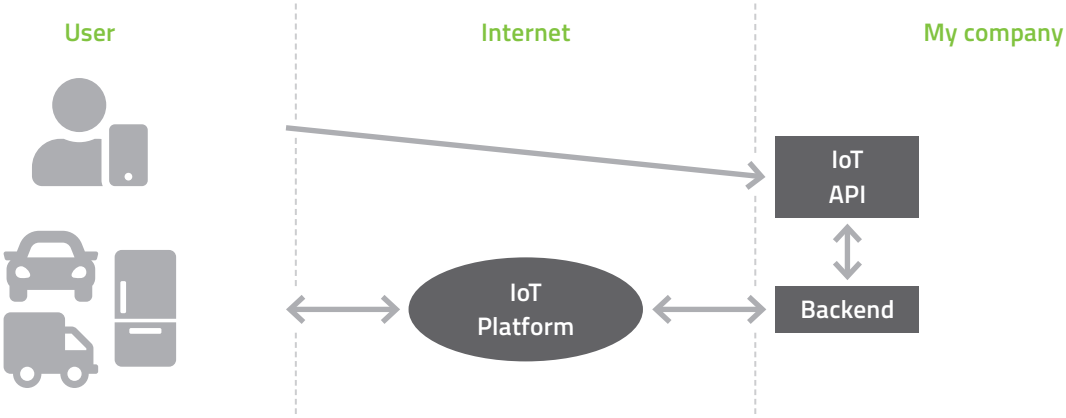
Im Zuge dieser enorm schnellen Verbreitung explodierte auch die Zahl der APIs. Sowohl für Nutzer als auch Entwickler wurde es unübersichtlich. Es brauchte Verfahren, die APIs zu verwalten, den Zugriff zu regeln und Ordnung in das Ganze zu bringen. Genau das leistet ein API-Management. Hier wird eindeutig definiert, wem welche Daten wie zur Verfügung gestellt werden. Ob die Anfrage vom Anwender selbst stammt oder eventuell von einer anderen API, spielt keine Rolle. Ganz im Gegenteil ist dies der Grund, warum digitale Dienste nicht länger nur nebeneinander stehende Anwendungen sind, sondern durch die Bereitstellung von APIs das Potential zu einem echten digitalen Ökosystem in sich tragen.

Ein anschauliches Beispiel dafür ist „If This Then That" (IFTTT). Dieser Dienst verbindet Ereignisse mit Aktionen nach einer einfachen Wenn-dann-Logik. Auf diese Weise verbindet er Lösungen von Anbietern, ohne dass diese überhaupt voneinander wissen. So kann ein digitales Türschloss das Ab- oder Aufschließen über eine API als Ereignis bereitstellen. Die Alarmanlage eines anderen Herstellers mag – ebenfalls über eine API – bestimmte Aktionen (scharfschalten, deaktivieren) ausführen. Über zwei einfache Regeln kann IFTTT diese beiden Geräte nun auf einer höheren Ebene integrieren: Wenn das Türschloss abgesperrt wird, aktivere die Alarmanlage; wenn es entsperrt wird, deaktiviere sie. Hierfür gibt es keine Geschäftsbeziehung zwischen dem Anbieter des Türschlosses und der Alarmanlage – wahrscheinlich wissen die beiden nicht mal etwas voneinander.

Ein anderes Beispiel für die Kombination unterschiedlicher Dienste zu etwas gänzlich Neuem sind digitale Assistenten wie Alexa, Siri und Co. Sie ermöglichen die Wiedergabe der aktuellen Top Ten über abonnierte Streamingdienste bis zur sprachgesteuerten Beleuchtungs- und Klimatechnik. Mit Hilfe solcher Dienste können sich Anwender eine digitale Umwelt ganz nach den eigenen Bedürfnissen schaffen. Anbieter auf der anderen Seite profitieren von einer deutlich engeren Kundenbindung. Denn während man eine solitäre App ohne großen Aufwand durch eine andere ersetzen kann, steigt der Aufwand mit zunehmender Vernetzung.

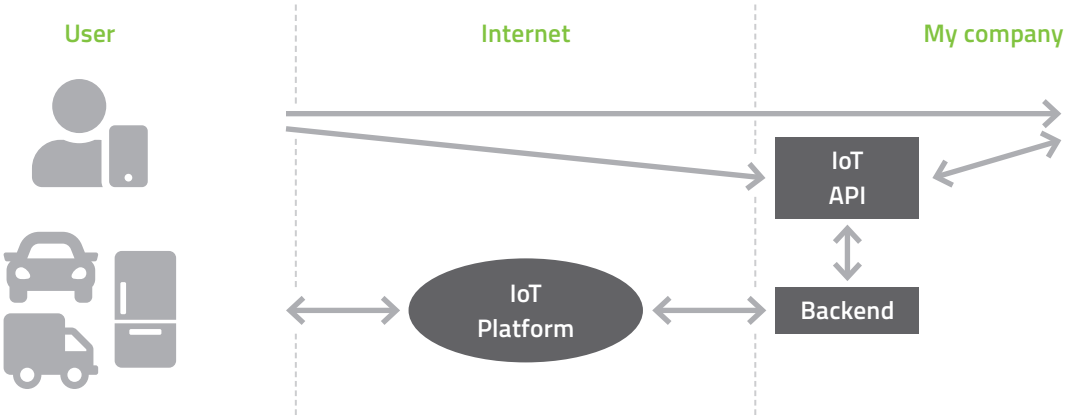
CIAM kommt ins Spiel

Viele APIs funktionieren auch ohne Identitätsprüfung problemlos. So ist bei der Abfrage von öffentlichen Informationen wie Öffnungszeiten, Tankstellenanzeige nach Benzinpreis oder Verspätungsmeldungen bei Bus und Bahn eine persönliche Anmeldung meist nicht erforderlich. Hier genügt die Bereitstellung einer API, die über ein Backend mit den IoT-Geräten kommuniziert. Die Identität des Anwenders spielt auf dieser ersten Evolutionsstufe des digitalen Ökosystems noch keine Rolle. Allerdings stößt dieses Vorgehen rasch an seine Grenzen. Wenn etwa mehrere Personen ein Auto nutzen, wird selbst bei der Abfrage von öffentlichen Informationen schnell der Wunsch nach eigenen Profilen aufkommen. Nicht aus Gründen der Datensicherheit, sondern wegen des Bedienungskomforts.



Einfaches API-Szenario ohne CIAM

Anders sieht es aus, wenn kostenpflichtige Zusatzdienste in Anspruch genommen werden. Zum Beispiel, wenn die Waschmaschine selbstständig Waschmittel nachbestellen soll. Oder auch bei der geteilten Nutzung von Geräten. So ist es durchaus sinnvoll, bei der gemeinsamen PKW-Nutzung die hinterlegten Social Media Profile oder Newspräferenzen nicht zu teilen.

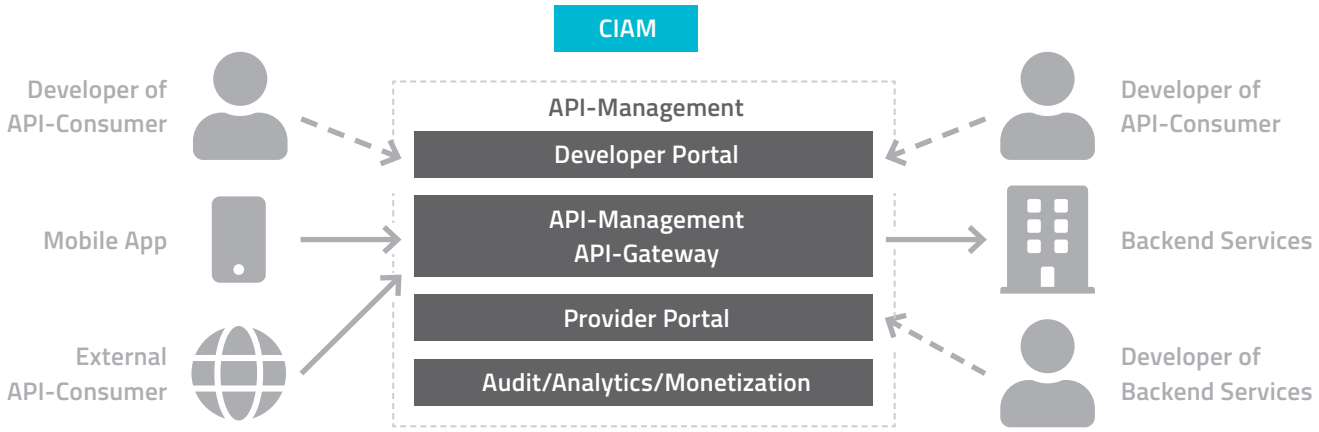


Auch wenn mehrere Nutzer mit individuellen Profilen gemeinsam IoT-Geräte bedienen, ist ein Identitätsmanagement erforderlich.

Die bisher besprochenen Szenarien lassen sich auch noch ohne API-Management umsetzen. Wenn aber zusätzliche Dienste mit eigenen APIs ins Spiel kommen, wird es sehr schnell unübersichtlich. Mit jeder weiteren API nimmt der Verwaltungs-, Konfigurations- und Integrationsaufwand exponentiell zu. Ganz zu schweigen von den damit zusammenhängenden Sicherheitsfragen. Der Überblick droht verloren zu gehen. Ein API-Management stellt diesen Überblick wieder her und vereinfacht den Aufbau eines digitalen Ökosystems für Entwickler enorm. Hier erhalten Entwickler alle erforderlichen Informationen für den Aufruf der API. Dabei geht es nicht alleine um die Dokumentation der Zugriffsmethoden, sondern auch um Beantragung, Freischaltung und Passwörter für den Zugriff auf die APIs. Denn natürlich soll keineswegs der Zugriff auf alle APIs aller Anwender ermöglicht werden. Das dafür nötige Einverständnis des Endanwenders wird durch das CIAM eingeholt und verwaltet.

Ein Blick hinter die Kulissen – Aufbau und Bestandteile eines API-Managements

API-Management – Components and Roles



Um die Funktion eines API-Managements zu verstehen, hilft es, die beteiligten Komponenten näher zu betrachten.

Da ist zunächst einmal das API-Gateway. Es stellt die Backend Services des Anbieters sowohl Apps als auch anderen API-Konsumenten abgesichert zur Verfügung.

Für den Anbieter besonders wichtig ist das Provider Portal. Es bietet den Entwicklern von Backend Services Funktionen zur Bereitstellung ihrer Dienste. Hier werden die Backend-Services mit der API verbunden und bei Bedarf Nutzungskosten festgelegt. Darüber hinaus lässt sich hier der Umfang der API-Nutzung durch die Entwickler verfolgen und Einblicke in die daraus eventuell erzielten Einnahmen gewinnen.

Für die Nutzer der APIs, also in erster Linie App-Entwickler, ist das Developer Portal der zentrale Anlaufpunkt. Hier erfahren sie, welche APIs zur Verfügung stehen und wie sie aufgerufen werden. Darüber hinaus können App-Entwickler hier ihre Apps für den Zugriff auf die API registrieren. Natürlich stellt sich dabei die Frage nach den Zugriffsrechten auf das Developer Portal. Die Erfahrung zeigt, dass die meisten Developer Portals als unternehmensinterne Lösung starten. Einfach aus dem Wunsch heraus, bereits entwickelte Tools den eigenen Mitarbeitern zur Verfügung zu stellen und Mehrfachentwicklungen der gleichen Funktionalität zu vermeiden. Relativ schnell zeigt sich dann, dass auch Partnerunternehmen und Zulieferer vom Developer Portal profitieren. Tatsächlich empfiehlt sich die Öffnung auch unabhängigen Dritten gegenüber. Dazu weiter unten mehr.

Aber ist es nicht riskant, völlig unbekannten Personen Zugriff auf unternehmenseigene Funktionen zu bieten? Jede ins Internet exponierte API erhöht die Angriffsfläche – aber dies ist auch ohne API-Management und Bereitstellung für Dritte so. Apps lassen sich dekompilieren oder belauschen, um die Funktionsweise einer API zu ermitteln. Der Aufruf einer API lässt sich mit vertretbarem Aufwand nicht zuverlässig verhindern. Statt

unnötig Energie in Nutzungsbeschränkungen zu investieren, ist es meist viel effizienter, Regeln für die Nutzung zu definieren und eine Dokumentation zu hinterlegen. Außerdem muss bei der Bereitstellung einer API – egal ob öffentlich oder nur für eine bestimmte App – eines klar sein: Alle Sicherheitskontrollen erfolgen im Backend. Keine App gilt jemals als vertrauenswürdig.

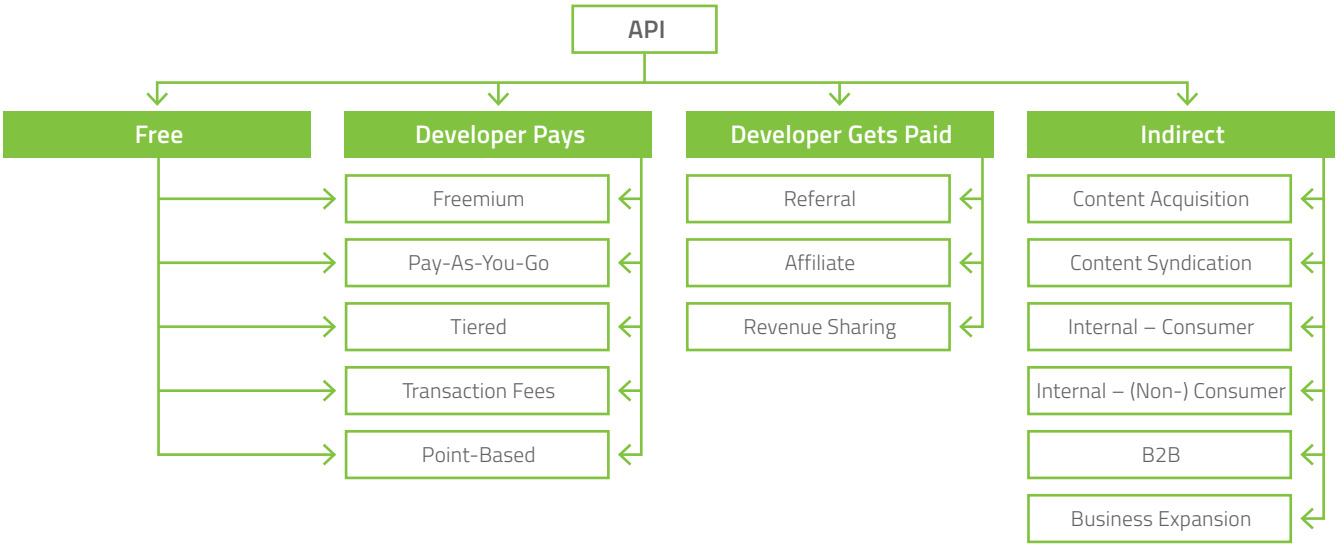
Wenn man so will, kann man auch sagen: Statt durch das Aufziehen hoher Mauern den Blick darüber nur umso interessanter zu machen, wäre es sinnvoller, eine Wegeordnung aufzustellen und gegebenenfalls Eintrittsgeld für die unmittelbare Besichtigung zu verlangen.

Auch wenn Monetarisierung derzeit noch kein allzu großes Thema ist, darf es bei der Betrachtung eines API-Managements nicht fehlen. Denn prinzipiell sind APIs ein ideales Werkzeug, um die digitalen Assets eines Unternehmens nutzbar zu machen und neue Geschäftsmodelle zu etablieren. Zuerst sollten jedoch drei grundlegende Fragen geklärt werden:

- Welchen Nutzen bietet die API möglichen Kunden?
- Wer zählt zu den möglichen Kunden (Partner, Unternehmen, unabhängige Entwickler)?
- Wie können sie aus der API selbst den größtmöglichen Nutzen ziehen?

Vor allem der letzte Punkt ist entscheidend für die Wahl des passenden Monetarisierungsmodells: kostenlos, Entwickler zahlt, Entwickler wird bezahlt oder indirekte Modelle.

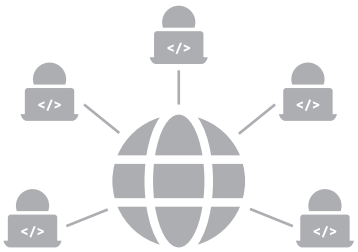
API Monetization Models and Variations



Je nach digitalem Asset bieten sich unterschiedliche Vermarktungsmöglichkeiten an. Freemium-Modelle, also die kostenlose Bereitstellung von Grundfunktionen, erlauben es den Kunden, sich vom Nutzen der Anwendung zu überzeugen. Pay-as-you-go eignet sich immer dann, wenn die API unregelmäßig und nur bei Bedarf genutzt wird. Tiered

Modelle ähneln Volumentarifen, bei denen der Kunde zum Beispiel eine bestimmte Zahl an API-Aufrufen pro Monat bestellt. Bei Überschreiten des Kontingents kann der Zugriff blockiert, der Preis erhöht oder ein zusätzliches Paket gebucht werden. Der Point-based Ansatz funktioniert ähnlich. Der Kunde erwirbt eine bestimmte Anzahl an Punkten, die als eine Art Währung für API-Aufrufe gelten. Auf diese Weise lassen sich verschiedene Operationen flexibel bepreisen. So könnte eine GET-Operation einen Punkt kosten, eine POST-Operation dagegen zwei Punkte. Ähnlich flexibel lassen sich auch die anderen Monetarisierungsvarianten nutzen, bei denen Entwickler z. B. über Affiliate-Programme entlohnt werden oder unterschiedliche Kostenstellen innerhalb eines Unternehmens für die Nutzung zahlen.

Egal, für welche Option man sich letzten Endes entscheidet: Das Modell muss zum Produkt passen. Es sollte transparent und leicht zu verstehen sein. Und es muss durch entsprechende Monetarisierungsfunktionen im API-Management unterstützt werden. Dazu gehört die flexible Definition von Regeln für erlaubte Transaktionen. Monitoring-Funktionen überwachen die einzelnen Aktionen und achten darauf, dass keine Limits überschritten werden. Und nicht zuletzt gilt es, die Kunden bestmöglich zu unterstützen. Zum Beispiel über das Developer Portal, dass ihnen die benötigten Werkzeuge und Informationen bereitstellt.



Wer den Entwicklern den Zugriff auf eigene Dienste eröffnet, multipliziert seine Chancen.

Auf zu neuen Zielgruppen – Öffnung der APIs für unabhängige Entwickler

Im Zusammenhang mit dem Developer Portal wurde schon darauf hingewiesen, dass auch Dritte Zugriff darauf haben können. Tatsächlich verbirgt sich dahinter ein strategischer Vorteil des API-Managements. Dazu muss man sich nur vergegenwärtigen, wie viele Entwickler es weltweit gibt. Wer diesen Entwicklern den Zugriff auf eigene Dienste eröffnet, multipliziert seine Chancen, Teil des digitalen Ökosystems der Anwender zu werden und obendrein neue Zielgruppen zu erschließen. Ohne Entwicklungskosten können so Lösungen entstehen, an die im eigenen Unternehmen noch gar nicht gedacht wurde. Auch internationale Roll-outs lassen sich mit flexiblen, lokalen Entwicklerteams oft effizienter durchführen, als mit einer großen Zentrale, die die Gegebenheiten vor Ort nicht kennt. Dass die vorgesehenen Regeln eingehalten werden, dafür sorgt das API-Management.

CIAM und API-Management – das neue digitale Dream Team

CIAM-Lösungen und API-Management sind nicht zwingend aufeinander angewiesen. Sie lassen sich auch losgelöst voneinander betreiben. Da aber immer mehr Anwendungsfälle in der einen oder anderen Form digitale Identitäten betreffen, ergänzen sich beide Ansätze in fantastischer Weise.

So muss sich das API-Management nicht selbst um die Identitätsklärung kümmern. Da APIs im Kontext des Kunden ausgeführt werden, ist dies auch gar nicht erforderlich. Eine API zur Abfrage von Börsenkursen ist an sich unkritisch. Sicherheitsrelevant wird es erst, wenn etwa die Wertstellung eines persönlichen Portfolios ermittelt werden soll. Das dafür nötige Identity and Access Management muss jedoch nicht von der API geleistet werden, sondern kann davon losgelöst erfolgen. Zudem werden APIs meist nicht alleine angeboten, sondern laufen parallel zu Web-Anwendungen, eCommerce-Systemen etc., die sowieso ein Access Management benötigen. Hinzu kommt, dass heutzutage die meisten Systeme verteilt aufgestellt sind – sowohl on-premise als auch bei verschiedenen

Cloud-Anbietern. Mit einem zentralen CIAM und einheitlicher Token-Vergabe, lassen sich die APIs auch in unterschiedlichen Welten komfortabel nutzen. Andernfalls muss sich der App-Entwickler unter Umständen mehrfach in verschiedenen API-Management-Systemen mit unterschiedlichen Regeln registrieren. Schlimmstenfalls kann dies dazu führen, dass sich der Endanwender mehrfach einloggen muss, um eine App zu bedienen. Mit der Unterstützung von Social Logins und eigenständigen Identity Providern gestaltet sich dies mit eigenem CIAM deutlich einfacher und anwenderfreundlicher. Gleiches gilt für die zentrale Lösung aller Aspekte rund um die Einwilligungen des Anwenders (User Consent).

Umgekehrt kann das API-Management das CIAM ebenfalls sehr wirksam unterstützen. Dies beginnt mit der Integration von Anwendungen. Mitunter laufen auf einem CIAM mehrere Dutzend, manchmal über 100 Anwendungen. Während CIAM-Systeme nicht immer gut in der effizienten Einbindung solcher Anwendungen sind, wurde das API-Management genau für diese Aufgabe geschaffen. Hinzu kommt das Application-Management, das einen kompakten Überblick über die vorhandenen Anwendungen und ihre Nutzung bietet.

Bereitgestellte Funktionalität lässt sich nachvollziehbar in Rechnung stellen.

Ein sehr interessanter Aspekt sind auch die Monetarisierungsfunktionen des API-Managements. Bieten sie doch die Möglichkeit, die Kosten des CIAMs an die Anwendungen weiterzuberechnen. Gerade in Konzernen, die das Prinzip der Fremdüblichkeit umsetzen müssen, lässt sich so die bereitgestellte Funktionalität nachvollziehbar in Rechnung zu stellen.

Gemeinsam stärker

In dieser kurzen Betrachtung haben wir gesehen, dass sich CIAM und API-Management auf vielen Ebenen geradezu ideal ergänzen. Beide Systeme vermögen die Effizienz des anderen nachhaltig zu steigern, Prozesse zu vereinfachen und darüber hinaus neue Geschäftsmodelle zu entwickeln und neue Zielgruppen zu erschließen. Es wird ohne Frage spannend bleiben, diese Entwicklungen weiter zu verfolgen.

3 Punkte, die Sie bei der Umsetzung Ihres API-Managements beachten sollten

Developer Portals

Gestalten Sie Ihr Developer Portal so intuitiv wie möglich. Sorgen Sie für eine umfassende, verständliche Dokumentation. Machen Sie die Nutzung für Dritte so einfach wie möglich. Je komfortabler der Zugriff auf die API ist, je mehr per Self-Service erreicht werden kann, umso mehr Entwickler werden sie nutzen.

Jenseits der EU-DSGVO

Bieten Sie Ihren Kunden mehr als die gesetzlich vorgeschriebene Compliance. Die Minimalumsetzung bietet jeder. Zusätzliche Transparenz und Privacy dagegen wird von den Kunden als Alleinstellungsmerkmal wahrgenommen.

Know-how & Failsafe to Use

Je einfacher und sicherer sich eine API nutzen (failsafe to use) lässt, umso schneller und intensiver wird sie Teil von digitalen Ökosystemen werden und dadurch neue Zielgruppen und Geschäftsmöglichkeiten eröffnen.