

Entlasten Sie Ihre SOA durch eine REST-Architektur mit APIs:

Der Grundstein für das moderne Fundament eines digitalen Business

01

EINFÜHRUNG

In den frühen 2000er-Jahren wurde das Zeitalter der serviceorientierten Architektur (SOA) eingeläutet, welcher man ein hohes Maß an Agilität zuschrieb. Die lose Verknüpfung und Verwendung von Standards führte jedoch nicht generell zu der von den Anwendern angestrebten geschäftlichen Transformation. Die Verwaltung der Technologie erwies sich als komplexer und anspruchsvoller, als es von den Experten erwartet wurde. Anders verhält es sich mit der RESTful API, der nächsten Entwicklungsstufe nach SOAP. Mit ihr ist eine flexible Anwendungsintegration möglich, da die vorausgesetzten Fertigkeiten auf breite Akzeptanz stoßen. Dadurch ermöglicht sie eine echte digitale Transformation des Kundennutzens, aber auch des Geschäfts an sich.

Unternehmen mit einer SOA (serviceorientierten Architektur) können nun den Übergang zu einer API-basierten Architektur vollziehen und von der verbesserten Konnektivität und einfachen Bedienung profitieren.

Die Kluft zwischen SOA (und Alt-Anwendungen) und API mag unüberwindbar erscheinen, doch mit den richtigen Praktiken und Plattformen ist der Weg von der SOA zur API-basierten Architektur ein durchaus sinnvolles Unterfangen. Dieses Paper beschreibt, wie die Umwandlung von SOAP-Webservices hin zu einem Teil eines breiteren produktiven API-Ökosystems gelingt.

02

SOA: eine revolutionäre Idee, die ins Stocken geriet

03

Digitale Transformation =Wandel des Kundennutzens

04

Der Unterschied zwischen SOA und APIs

Der Übergang zu einem API-gestützten Kundennutzen

05

Der Prozess im Überblick

06

5 Best Practices in Bezug auf die API

07

Die Rolle der API-Management-Plattform

08

Fazit



SOA: eine revolutionäre Idee, die ins Stocken geriet

In den 1990er-Jahren erfreute sich das Internet zunehmender Beliebtheit. Es demonstrierte dabei die beispiellose Marktstärke offener Standards wie HTML und HTTP. Gleichzeitig fragten sich die Menschen, ob dieser Effekt auch mit Software an sich zu erreichen sei. Könnten Anwendungen offene Standards nutzen, um Barrieren für Integration und übergreifende Zusammenarbeit abzubauen? Könnte dies auf die gleiche Art und Weise gelingen wie bei den Webstandards, mit denen das Internet aus dem Labor hinaus in die Wirtschaft katapultiert wurde?

Software-Architekten stellten sich die Frage, wie schwer es wäre, Software zu etablieren, die Funktionen aus anderen Anwendungen aufrufen könnte. Diese sollte genauso einfach wie Verbraucher von einer Website zur anderen springen. Wie wäre es beispielsweise mit einem Finanzportal für Verbraucher, das verschiedene Börsenticker-Feeds nutzen könnte – ohne die kosten- und zeitintensive individuelle Codierung, die damals erforderlich war? Bis zu diesem Zeitpunkt waren bei der Verbindung von Anwendungen proprietäre Middleware und benutzerdefinierte Links verwendet worden. Diese benötigten spezifische Firewall-Konfigurationen, um zu funktionieren.

Zu Beginn des neuen Jahrtausends überraschten die größten Technologieunternehmen die Branche mit dem Vorhaben, gemeinsam an offenen Standards arbeiten zu wollen. Diese sollten eine kostenlose, internetbasierte Integration von Anwendungen begünstigen. Diese Standards ermöglichten die Realisierung einer „serviceorientierten Architektur“ oder von SOA-Standards. Sie basierten auf der XML-Sprache und beinhalteten unter anderem das Simple Object Access Protocol (SOAP) und die Web Service Description Language (WSDL). Diese Standards bildeten die Grundlage für bedeutende neue Plattformen wie Microsoft.NET, BEA (später Oracle) WebLogic, die neue Generation von IBM WebSphere Servern, SAP NetWeaver und Oracle Fusion Middleware.

Das SOA-Konzept war umsetzbar geworden. Da alle Branchenriesen an Bord waren, riefen viele Unternehmen ehrgeizige Programme zum Aufbau serviceorientierter Architekturen ins Leben. Dabei wurde die Softwarefunktionalität in Form von SOAP-basierten Webservices verfügbar gemacht. Diese ließen sich ganz einfach über HTTP innerhalb und außerhalb des Unternehmens verbinden. Es entstanden neue Konstrukte wie „Enterprise Service Bus“ (ESB), die es den Unternehmen ermöglichten, ihre Webservices in Softwarestrukturen zu platzieren, die sich leicht in neue Anwendungen integrieren ließen.

Die serviceorientierte Architektur funktionierte bis zu einem gewissen Grad. Viele Erfolge boten den Anwendern ein bisher selten erreichtes Maß an Komfort. Wichtige Anwendungen für ERP und Logistik veröffentlichten Neuauflagen mit Hunderten Funktionen, die als Webservices bereitgestellt wurden. Entwickler entdeckten eine neue Agilität beim Verknüpfen von Anwendungen und dem Erschaffen neuer Benutzererfahrungen. Neue „Composite Apps“, die mithilfe von Webservices erstellt wurden, ermöglichten eine bessere Anpassung an die Anforderungen der Benutzer.

Doch schon nach einigen Jahren geriet das SOA-Experiment ins Stocken. Was die Offenheit und ungewöhnliche Zusammenarbeit konkurrierender Technologieanbieter anging, war SOA zweifellos revolutionär. Doch bei der Anwendung blieb es eine umständliche Technologie. Die ESB-Netzwerkarchitektur erwies sich als teuer und kompliziert in der Wartung. Man war davon ausgegangen, den Geschäftswert nach Abschluss des SOA-Projekts zu erhalten. Jedoch schien die Arbeit kein Ende zu nehmen. Gut strukturierte SOAs mit intuitiven Verzeichnissen von Webservices, die in mehreren Anwendungen verwendet werden konnten, sollten den Geschäftswert einbringen. Dieser war aber nur schwer zu erreichen.

Stattdessen entwickelte sich im ersten Jahrzehnt des 21. Jahrhunderts eine ausgesprochen heterogene Umgebung. Dazu gehörten organisierte SOA-Programme, eine ganze Reihe von ESB-orientierten Architekturen und viele Unternehmen mit verstreuten, schlecht organisierten Webservices. Nicht wenige jedoch hinkten hinter der SOA-Entwicklung hinterher.

So war ein neuer, einfacherer Ansatz erforderlich.



Digitale Transformation = Wandel des Kundennutzens

Die SOA war in gewisser Hinsicht ein Opfer ihres eigenen Erfolgs geworden.

Doch trotz aller Fehler zeigte SOA eindeutig, dass denjenigen, die mit Technologie das eigene Business transformieren könnten, ein immenser wirtschaftlicher Gewinn winkte.

Die SOA öffnete den Führungskräften die Augen für das Potenzial von Composite Applications. Diese könnten für die Rolle des Benutzers relevante Services beinhalten.

Sie sahen die Verheißung von lose verknüpften Anwendungen, um den Kundennutzen radikal zu erneuern und zu verbessern. Nun ging es darum, die Technologie zu entwickeln, die einen solchen Übergang möglich machen würde.

Die Kundenbindung über alle Kanäle hinweg ist ein Beispiel für eine gelungene Transformation des digitalen Business. In einer Bank könnte man den Kunden beispielsweise nahtlos mit Kontoinformationen, Transaktionen, Dritten wie Anlagekonten, Kreditdienstleistungen und Kundendienstmitarbeitern über Telefon, PC, Smartphone, Bankschalter in der Filiale usw. verbinden. Dazu müssen verschiedene Anwendungen einfache und agile Verbindungen untereinander herstellen. Die serviceorientierte Architektur war für diese Vision zu klobig.

Die Lösung erschien in Form von standardbasierten Programmierschnittstellen (Application Programming Interfaces, APIs). Die API, eine Softwareeinheit, die Anwendungen und Hardware miteinander verknüpfte, war alles andere als neu. Aber durch das Hinzufügen

einfacher, offener Standards wurde sie zum Vehikel einer echten IT-Revolution. Wenn man in der IT-Fachliteratur heute über APIs diskutiert, wird stets der Bezug zum Architekturstil Representational State Transfer (REST) und dem leichtgewichtigen Datenformat JSON hergestellt. Derartige RESTful APIs bieten Entwicklern und Architekten eine besonders einfache Konnektivität. Ganz so, wie man sie sich im Zusammenhang mit den SOA-Standards vorgestellt hatte, aber niemals realisieren konnte.

RESTful APIs, die Netzwerkprotokolle wie HTTP verwenden, können beide Softwarekomponenten bemerkenswert einfach miteinander verbinden. Die vereinbarten Standards für RESTful APIs enthalten einige einfache Methoden wie GET und PUT (Daten).

Im Bereich der Bank konnten Entwickler mit relativ geringem Aufwand eine Banking-App für den Verbraucher mit Zugang zu Konten, Handelssystemen usw. zusammenstellen. Dazu benötigten sie lediglich die Adressen (URLs) und Datenparameter der APIs. Zudem gestaltete sich die Änderung der App und der zugrunde liegenden API-Verbindungen ziemlich einfach. Auf diese Weise wurden APIs zu den Werkzeugen der digitalen Transformation.





Der Unterschied zwischen SOA und APIs

APIs versuchen zwar, ähnliche Vorteile zu erzielen, im Gegensatz zur SOA ist ihr Hauptmerkmal jedoch die Einfachheit.

Diese war auch entscheidend, um Mobilgeräte zur Interaktion mit anderen Systemen zu befähigen. Die SOA war nie mit einer derartigen Einfachheit entworfen worden. Daher war die Nutzung von SOA außerhalb des Unternehmens, z. B. mit Partnern usw., eingeschränkt. In Bezug auf Ressourcen ist die API beispielsweise über eine URL mit der Standard-Webtechnologie HTTP aufrufbar. Der API-Ansatz bietet die Möglichkeit, klein anzufangen. Eine begrenzte Basisfunktion lässt sich schnell hin zu komplexeren Anwendungsfällen ausbauen.

Der Übergang zu einem API-gestützten Kundennutzen

Das Motto „Lean-Start-up“ von „Fail-Fast“ beschreibt treffend die Startphase mit einem API-Programm.

In gewisser Weise schlüpfen Sie dabei in die Rolle eines Start-ups, das mitten im Business steckt. Sie werden Technologien entwickeln, die bisher getrennte Systeme und Geschäftsprozesse auf neuartige Weise miteinander verbinden – genauso wie ein Start-up. Vielleicht sind Sie sogar ein kleiner Störfaktor, mit allen Vor- und Nachteilen, die dieser mit sich bringt.

Klein anfangen. Fehler frühzeitig erkennen.

Sie müssen keine überdimensionierte API-Initiative für das ganze Unternehmen starten. Dies könnte zu echten Problemen führen. Für eine schnelle Wertschöpfung ist es besser, ausgehend von den Geschäftszielen nach außen zu denken.

Beginnen Sie beim Kunden. Bei den digitalen Geschäftsinteraktionen von heute gibt es eine ständig wachsende Anzahl von Touchpoints. Überlegen Sie einmal, wie der Kundennutzen von einer Überquerung dieser Touchpoints profitieren könnte.

Kann beispielsweise eine Kundeninteraktion komplikationslos vom Desktop zum Tablet oder Handy und zurück bewegt werden? Falls nicht, so ist das ein hervorragender Ausgangspunkt, um über APIs nachzudenken und darüber, wie sie den Kundennutzen über Touchpoints verbessern. Die API-Technologie stellt die Grundlage für die Verbindung von Systemen und Touchpoints dar. Sie verbindet zudem die Menschen, auf die der Kundennutzen zugeschnitten ist.

Der Prozess im Überblick

Sobald Sie einen durch APIs zu verbessernden Kundennutzen festgestellt haben, ist es Zeit, loszulegen. Der erste Schritt besteht darin, ein Modell des Prozessablaufs zu erstellen. Es soll aufzeigen, was in jedem Schritt der Kundeninteraktion mit verschiedenen zugrunde liegenden Systemen passieren wird. Dieser Flow kann angesichts der Funktionsweise digitaler Unternehmen etwas komplex ausfallen.





So kann beispielsweise der erste Touchpoint für den Kunden über einen Webbrowser erfolgen, der einen E-Commerce-Shop aufruft. Der nächste Touchpoint könnte eine mobile App sein, die den Transaktionsstatus aus einer Fulfillment- und Logistikanwendung bezieht. Ein dritter Touchpoint könnte über das Handy und ein Interactive-Voice-Response-System (IVR) erfolgen, das Daten zum Auftragsstatus einholt aus einem Antrag auf Frachtabfertigung bei einem Dritten, z. B. dem United States Postal Service.

Wie **Abbildung 1** zeigt, muss jeder Touchpoint (und die dazugehörige Anwendung) als eine API bereitgestellt werden. Einige dieser Anwendungsfunktionen sind möglicherweise bereits als SOAP-Webservice verfügbar. In diesem Fall ist es notwendig, eine sogenannte SOAP-to-API Conversion durchzuführen. Viele API-Management-Tools stellen diesen Prozess standardmäßig zur Verfügung.

Dabei wird der SOAP-Webservice zu einer RESTful-API umgewandelt.

Jede API benötigt einen „API-Konsumenten“, bei dem es sich typischerweise um ein Stück des Kundencodes handelt. In unserem Beispiel muss die mobile App zur Bestellverfolgung die API, die in der Fulfillment-Anwendung bereitgestellt wird, „konsumieren“.

5 Best Practices in Bezug auf die API

Der Übergang von SOAP-Webservices und Alt-Anwendungen hin zu einem API-gesteuerten Kundennutzen funktioniert besser, wenn Ihr Unternehmen bewährte Best Practices verwendet. Dafür sind eine detaillierte Planung, die Unterstützung durch das Management, ein gewisses Budget, klare Verantwortlichkeiten, vereinbarte Aufgaben, Zeitpläne usw. erforderlich. Verfügen Sie nicht über derartige grundlegende Projektmanagement-Praktiken, werden die APIs Ihnen



Abbildung 1: Verschiedene Touchpoints erfordern mitunter APIs auf mehreren zugrunde liegenden Systemen.



Schwierigkeiten bereiten.

In den letzten Jahren sind viele Best Practices entstanden, die den SOA-to-API-Übergangsprozess bestmöglich funktionieren lassen. Zu den wichtigsten Aspekten zählen:

01

Mapping des Kundenverhaltens

Damit APIs relevant sind, müssen sie auf sich ändernde Kundenanforderungen abgestimmt sein und sich diesen anpassen können. Dies kann beispielsweise bedeuten, dass zusätzliche, für den Kunden wertvolle Informationen im Voraus bedacht werden. Es ist entscheidend, die Browsing-Muster zu verstehen.

02

Verständnis der Unterschiede bei der Nutzung von Webservices oder APIs

Die Implementierung von SOA führte, entsprechend der Granularität des Designs, oft zu Hunderten oder Tausenden Webservices. APIs sollten die Aufgabe des sie verwendenden Entwicklers erleichtern. Daher werden Webservices oft in einem einzigen API-Aufruf kombiniert. Ein Entwickler, der eine App schreibt, interessiert sich beispielsweise nicht für die vielen Schritte, die für einen Kauf oder einen neuen Kontakt nötig sind. Eine API-Methode, die alle zur Erfüllung der Aufgabe erforderlichen Webservices verwendet, vereinfacht die Aufgabe des Entwicklers.

03

Planung des Übergangs der APIs von isolierten Teams hin zu einer unternehmensweiten Akzeptanz

Wenn Sie APIs richtig angehen, schaffen Sie es natürlich auch über die „Start-up“-Phase hinaus. Eines Tages werden Sie in Ihrem Unternehmen möglichst vielen Menschen die Verwendung von APIs ermöglichen wollen. Immerhin sind APIs sowohl ein Business-Tool (oder mehr) als auch ein Bestandteil der IT. Für die Ausweitung auf das gesamte Unternehmen bedarf es einer guten Planung.

04

Verwaltung der API-Kontrolle im gesamten Unternehmen

Eine zunehmende Verwendung von APIs sollte keinen Kontrollverlust nach sich ziehen. Eine strenge Kontrolle der APIs ist entscheidend für deren effektive Eingliederung in unterschiedliche Projekte und verschiedene Unternehmenseinheiten. Für eine gesteigerte Effizienz sollten die APIs einheitlich gestaltet sein. Der beste Ansatz für eine erfolgreiche Kontrolle ist die Einführung einer unternehmensweiten API-Plattform.

05

Ausweitung der Entwicklung von APIs auf einen größeren Entwicklungspool

Erfolgreiche API-Programme beinhalten ein API-Entwicklerportal. Das Portal bietet die Möglichkeit, mit Ihrem ursprünglichen Team von API-Experten andere Personen im Unternehmen zu unterstützen – sowohl API-Ersteller als auch -Konsumenten. Dazu gehört auch die Bereitstellung eines Katalogs mit den bestehenden APIs. So werden die Entwickler dazu befähigt, existierende APIs testen, um deren Verwendung zu erlernen.

Community-Foren bieten einen Ort zum Peer-to-Peer-Lernen usw. Es bleibt abzuwarten, wie die Menschen APIs verwenden werden, sobald sie ihr Potenzial verstehen. Mitunter kommt es zu ungeplanten Innovationen, wenn Fachleute, die nicht der IT-Abteilung angehören, mit den bestehenden APIs Tools entwickeln, die sich auf ihre geschäftlichen Herausforderungen beziehen.





Die Rolle der API-Management-Plattform

Eine API-Management-Plattform bietet dem Unternehmen die Möglichkeit, den API-Lebenszyklus vom Aufbau bis zur Außerbetriebnahme zu optimieren. Sie können Anpassungen durchführen, wie die SOAP-to-API Conversion. Dies erleichtert den Aufbau von APIs aus Ihren bestehenden Diensten. Die Plattform trägt dazu bei, die API-Nutzung mit Zuverlässigkeit, Sicherheit und Skalierbarkeit zu beschleunigen. Mit ihr können APIs von der Planung, Entwicklung und Bereitstellung bis hin zur Außerbetriebnahme einheitlich verwendet werden, was den Kundennutzen verbessern soll.

Das ursprünglich einfach konstruierte API-Management entwickelt sich ständig weiter, da es mit der Komplexität und dem Umfang der immer anspruchsvolleren API-Strategien Schritt halten muss. Tatsächlich leisten die

aktuellen Plattformen oft mehr als nur die Verwaltung des API-Lebenszyklus. Wie **Abbildung 2** zeigt, können die Plattformen ziemlich vielseitig sein. So umfassen sie zum Beispiel neben RESTful-APIs nun auch routinemäßig Microservices.

Sie helfen bei der Verwendung von APIs und Microservices in der Anwendungsentwicklung, bei B2B-Anwendungsfällen sowie bei der Content Collaboration. So können sie den Prozess erleichtern, bis die APIs unter den Partnern funktionieren. Sie ermöglichen zudem eine erweiterte Auswertung der API- und Microservice-Leistungen und -Abhängigkeiten.

Da APIs in der Lage sind, die Integration zu optimieren, ist das API-Management oft eine zentrale Komponente einer hybriden Integrationsplattform.

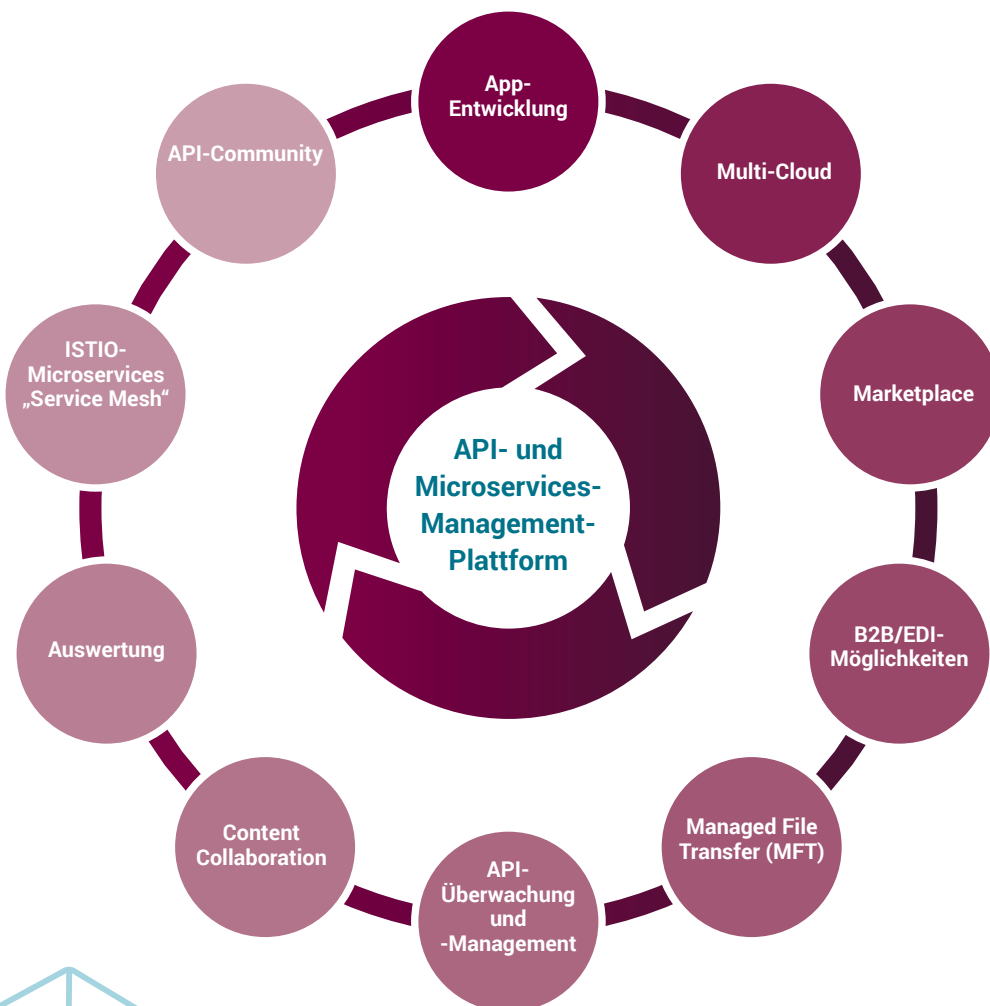


Abbildung 2: Die neue Generation von API- und Microservices-Management-Plattformen umfasst eine Vielzahl von Funktionen, darunter Anwendungsentwicklung, Content Collaboration und Auswertung.





FAZIT

Unternehmen mit SOAP-Webservices und serviceorientierter Architektur können den Übergang wagen und die Vorteile von RESTful APIs nutzen.

Dies führt sie auf den Weg zu transformativen Verbesserungen des Kundennutzens und der digitalen Transformation.

Der Einsatz von APIs ermöglicht ein dynamisches digitales Business in einem Umfang, der erstmals mit der Einführung der serviceorientierten Architektur vorstellbar wurde. Der erste Schritt läutet Ihren Übergang hin zu einer API-basierten Architektur ein. Moderne API-Managementpläne und Best Practices unterstützen den Prozess. Zu den wichtigsten Faktoren gehören die Start-up-Denkweise und das „Fail-Fast“ beim Start im kleinen Rahmen.

Anschließend besteht die Herausforderung darin, den Zugriff auf APIs im gesamten Unternehmen auf beherrschbare Weise zu erweitern. Hier kann die API-Management-Plattform eine große Hilfe sein, denn sie ermöglicht die Erstellung von Entwicklerportalen und die konsistente Anwendung von Richtlinien zum API-Lebenszyklus. Die beste Voraussetzung für eine erfolgreiche API-gesteuerte Transformation des Kundennutzens ist die Kombination von Best Practices und einer API-Management-Plattform zur Gewährleistung von Konsistenz und Kontrolle.

APIs müssen integriert, automatisiert und allgegenwärtig sein. Die Management-Plattform muss ISTIO-Service-Mesh, Multi-Cloud-Bereitstellung, DevOps und CI/CD-Nativity unterstützen. Mit diesen Funktionen können APIs und Microservices als Grundlage für spannende neue Erfahrungen für Kunden und Partner dienen.

Wie sieht eine API-First-Strategie für digitales Business aus?

Erfahren Sie, wie Sie mit Ihrem Business die digitale Kluft überbrücken können.

[VIDEO ANSEHEN](#)

Verändern Sie den Kundennutzen und Ihr Business mit API-Management

Wie trägt API-Management zum Erfolg Ihres Geschäfts bei?

[MEHR ERFAHREN](#)

**Try AMPLIFY
API Management
online for free**

SIGN UP TODAY

axway.com/en/products/api-management

Copyright © Axway 2019. Alle Rechte vorbehalten.
axway_WP_give_your_SOA_a_REST_with_APIs:
The_cornerstone_of_a_modern
digital_business_foundation

