



---

## Inhalt

- 1 Überblick
  - 2 Die Integration hat sich verändert
  - 2 Die bisherige Reise – SOA, ESBs und APIs
  - 3 Argumente für eine agile Integrationsarchitektur
  - 3 Aspekt 1: Differenzierte Integrationsbereitstellung
  - 4 Aspekt 2: Dezentrale Integrationsverantwortung
  - 5 Aspekt 3: Cloudnative Integrationsinfrastruktur
  - 5 Wie haben sich moderne Integrations-Laufzeitmodule mit Blick auf die agile Integrationsarchitektur verändert?
  - 6 Agile Integrationsarchitektur für die Integrationsplattform
  - 6 Die IBM Cloud Integration-Plattform
- 

# Agile Integrationsarchitektur

*Containerbasierte und auf Mikroservices abgestimmte schlanke Integrations-Laufzeitmodule*

Eine digitale Transformation mit agiler Integration auf Basis eines gleichermaßen agilen Ansatzes bietet Ihnen die Möglichkeit, schnell die Anforderungen von Multicloud, Dezentralisierung und Mikroservices zu erfüllen.

## Überblick

Unternehmen, die eine digitale Transformation durchlaufen, müssen sich auf neue Wege zur Verwendung und Implementierung von Integrationstechnologien einlassen, damit sie schnell und angemessen nach den Zielen von Multicloud, Dezentralisierung und Mikroservices handeln können. Die Anwendungsintegrationsschicht muss transformiert werden, damit Unternehmen gezielt neue Kundenerlebnisse schaffen können, statt Architektur- und Entwicklungsmodelle zu verwenden, die einer Maximierung der Unternehmensproduktivität entgegenlaufen.

Viele Unternehmen haben agile Anwendungsverfahren wie die Mikroservices-Architektur eingeführt und profitieren jetzt von dieser Umstellung. Dieser Ansatz ergänzt und beschleunigt die API-Strategie eines Unternehmens. Unternehmen sollten diesen Ansatz möglichst auch nutzen, um die vorhandene ESB-Infrastruktur zu modernisieren und so effektivere Möglichkeiten zu schaffen, ihre Integrationservices in der Private oder Public Cloud zu verwalten und zu betreiben.

Diesem Whitepaper liegt ein [Buch](#) zugrunde, in dem die Vorzüge der sogenannten **agilen Integrationsarchitektur** untersucht werden. Dabei handelt es sich um einen containerbasierten, dezentralisierten und auf Mikroservices abgestimmten Ansatz für Integrationslösungen, der die Anforderungen der digitalen Transformation in Bezug auf Agilität, Skalierbarkeit und Ausfallsicherheit erfüllt.



## Integration hat sich verändert

Schätzungen von IDC zufolge geben Unternehmen für digitale Transformations-Initiativen in den nächsten fünf Jahren 20 Billionen US-Dollar aus.<sup>1</sup> Was steckt hinter der enormen Ausgabenexplosion? Das ständig wachsende Bedürfnis, neue Kundenerlebnisse durch integrierte Anwendungen zu schaffen, die Daten aller Art nutzen.

Keine leichte Aufgabe – Prozesse und Informationsquellen zur richtigen Zeit und im richtigen Kontext zusammenzuführen, ist nahezu unmöglich. Dies gilt umso mehr, wenn Sie die vermehrte Einführung von SaaS-Geschäftsanwendungen berücksichtigen. Neue Datenquellen müssen in die Geschäftsprozesse eingebunden werden und dienen als Differenzierungsmerkmal im Wettbewerb.

---

*„Um neue Kundenerlebnisse zu schaffen, müssen Unternehmen sich einer immer größer werdenden Anzahl von Anwendungen, Prozessen und Informationsquellen bedienen – die den Bedarf des Unternehmens an Integrationsmöglichkeiten und entsprechenden Investitionen stark ansteigen lassen.“*

---

## Der Nutzen der Anwendungsintegration für die digitale Transformation

Wenn Sie Ihre Agenda zur Schaffung neuer Kundenerlebnisse betrachten und sich darauf konzentrieren, wie der Zugriff auf Daten erfolgt und wie Daten für die Services und APIs, die diese Initiativen antreiben, bereitgestellt werden, erkennen Sie mehrere wesentliche Vorteile, die sich durch die Anwendungsintegration ergeben:

- Effektiver Umgang mit Disparität – Greifen Sie ungeachtet der Vielfalt Ihrer Multicloud-Umgebung auf Daten von jedem System aus und in jedem Format zu und stellen Sie Homogenität her.
- Know-how der Endpunkte – Moderne Integration umfasst Wissen zu komplexen Protokollen und Datenformaten, integriert aber auch Wissen über die tatsächlichen Objekte, die Geschäftstätigkeit und die Funktionen in den Endsystemen.

- Innovation durch Daten – Was Anwendungen innovativ macht, ist größtenteils die Möglichkeit, Daten über die Anwendungsgrenzen hinaus zu kombinieren und ihnen einen Sinn zu geben. Diese Eigenschaft wird insbesondere in der Mikroservices-Architektur offenbar.
- Unternehmens-relevante Basisfähigkeiten – Integrations-‘Flows’ ziehen einen großen Teil ihres Werts aus der darunterliegenden Laufzeitumgebung mit den Unternehmens-kritischen Funktionen für Fehlerbehebung, Fehlertoleranz, Log-Mechanismen, Performance Analyse usw.

Die Integrationsumgebung ändert sich, um mit den Datenverarbeitungsanforderungen der Unternehmen und Märkte Schritt zu halten. Doch wie hat sich die Entwicklung von SOA und ESBs zur modernen, containerisierten, agilen Integrationsarchitektur vollzogen?

## Die bisherige Reise – SOA, ESBs und APIs

Bevor wir uns mit der Zukunft der agilen Integration befassen, müssen wir ihre Ursprünge verstehen. SOA-Muster (SOA, serviceorientierte Architektur) tauchten erstmals zu Beginn des Jahrtausends auf. Die breite Akzeptanz der Standards, auf denen SOA basierte, versprach eine glänzende Zukunft, in der jedes System jedes andere System erkennen und mittels synchroner Bereitstellungsmuster mit jedem anderen System kommunizieren konnte.

Mit einem kurzen Sprung vorwärts landen wir mitten in der ESB-Bewegung (ESB, Enterprise Service Bus). Diese Technologie sollte die Verbindung mit Back-End-Systemen ermöglichen und basierte auf dem ‘Hub-and-Spoke’ Prinzip. Das ESB-Muster wurde in vielen Unternehmen erfolgreich implementiert. Im cloudnativen Bereich stößt es jedoch nicht auf viel Gegenliebe, da es als schwerfällig und nicht sehr agil gilt. Wie konnte es dazu kommen?

## Entscheidend sind einige wenige Fakten, die häufig in Wechselbeziehung zueinanderstehen:

- SOA war komplexer als eine reine ESB-Implementierung, vor allem was die Übernahme der Finanzierung für ein unternehmensweites Programm betraf.
- ESB-Implementierungen wurden oft als einzelne Infrastruktur für das gesamte Unternehmen umgesetzt, mit zehn, zwanzig, hundert, usw. Integrationen, die in einem zentralen Produktionsserver-Cluster installiert wurden. Obwohl das ESB-Muster keine starke Zentralisierung erfordert, war diese meist charakteristisch für die resultierenden Topologien.

<sup>1</sup>IDC MaturityScape Benchmark: Digital Transformation Worldwide, 2017, Shawn Fitzgerald, Golluscio.

- Die großen Einsparungen, die die Unternehmen sich von zentralen ESB-Mustern versprochen, blieben häufig aus, weil die Projektschnittstellen nicht auf andere Projekte übertragbar waren.
- Die Finanzierung von unternehmensübergreifenden Initiativen wie ESB erwies sich als schwierig und häufig waren die bereitgestellten Mittel auf wiederverwendbare Services beschränkt, mit denen sich zumindest die Erstellungskosten hereinholen ließen.

---

*Bei ESB-Mustern war es häufig problematisch, die Finanzierung unternehmensübergreifender Initiativen dauerhaft zu sichern, da diese sich nicht einer bestimmten Geschäftsinitiative zuordnen lassen.*

---

Dies hatte zur Folge, dass die Erstellung von Services durch ein spezialisiertes SOA-Team bei Projekten nicht zu dem erhofften Erfolgsmotor, sondern zu einem Engpass wurde. Das zentrale ESB-Muster bekam dadurch einen schlechten Ruf.

Bei der auf ESB-Muster angewandten serviceorientierten Architektur handelt es sich um eine unternehmensweite Initiative zur Erstellung wiederverwendbarer, synchron verfügbarer Services und APIs, die eine schnellere Erstellung neuer Anwendungen und die Integration von Daten aus anderen Systemen ermöglichen.

Die Mikroservices-Architektur hingegen ist eine Option, die die Erstellung einer einzelnen Anwendung mit Fokus auf Agilität, Skalierbarkeit und Ausfallsicherheit ermöglicht.

### **Argumente für eine agile Integrationsarchitektur**

Was hat Mikroservices-Konzepte im Anwendungsbereich so populär gemacht? Sie stellen einen alternativen Ansatz für die Strukturierung von Anwendungen dar. Eine Anwendung ist bei diesem Konzept keine monolithische Anwendung, die auf einem bestimmten Server ausgeführt wird, sondern eine Sammlung kleinerer Komponenten, die vollkommen unabhängig ausgeführt werden.

Die Mikroservices-Architektur hat drei entscheidende Vorteile:

1. Größere **Agilität** – Mikroservices sind so klein, dass sie isoliert zu verstehen sind und unabhängig voneinander ausgetauscht werden können.

2. Elastische **Skalierbarkeit** – Die Ressourcennutzung kann komplett dem Geschäftsmodell zugeordnet werden.
3. Diskrete **Ausfallsicherheit** – Bei entsprechender Entkopplung wirken sich Änderungen an einem Mikroservice zur Laufzeit nicht auf andere Mikroservices aus.

Wie wäre es angesichts dieser Vorteile, wenn wir die Integration, die in der Regel in zentralen Silos erfolgt, mit einer neuen Perspektive auf Basis einer Mikroservices-Architektur neu konzipieren würden? Das Ergebnis bezeichnen wir als „**agile Integrationsarchitektur**“.

---

*Eine agile Integrationsarchitektur ist definiert als „containerbasierte, dezentrale und auf Mikroservices abgestimmte Architektur für Integrationslösungen“.*

---

Für die agile Integrationsarchitektur sind drei verwandte Aspekte bestimmend:

#### **Aspekt 1: Differenzierte Integrationsbereitstellung.**

Welche Vorteile hätte es, wenn wir die Integrationen aus einem zentralen isolierten ESB auf mehrere leicht-gewichtige und unabhängige Module verteilen könnten?

#### **Aspekt 2: Dezentrale Integrationsverantwortung.**

Wie kann die Organisationsstruktur von einem verteilten Ansatz profitieren?

**Aspekt 3: Cloudnative Integrationsinfrastruktur.** Welche weiteren Vorteile lassen sich durch einen vollständig cloudnativen Integrationsansatz erzielen?

#### **Aspekt 1: Differenzierte Integrationsbereitstellung**

Die zentrale Bereitstellung eines Integrationshubs oder ESB-Musters, bei der alle Integrationen auf einem aufwendig gepflegten einzelnen Paar von Hochverfügbarkeitsservern bereitgestellt werden, führt nachweislich zu Projektengpässen. Bei jeder Bereitstellung auf den gemeinsam genutzten Servern besteht die Gefahr, dass vorhandene kritische Schnittstellen destabilisiert werden. Es ist nicht möglich, für ein Einzelprojekt ein Versionsupgrade der Integrationsmiddleware durchzuführen, um auf neue Features zugreifen zu können.

Die unternehmensweite ESB-Komponente könnte in kleinere, einfacher zu handhabende und dedizierte Teile aufgespalten werden, in einigen Fällen eventuell sogar soweit, dass pro bereitgestellter Schnittstelle ein Laufzeitmodul vorliegt. Diese Muster für eine „differenzierte Integrationsbereitstellung“ liefern spezialisierte Container passender Größe, die sich durch verbesserte Agilität, Skalierbarkeit und Ausfallsicherheit auszeichnen und sich stark von den zentralen ESB-Mustern der Vergangenheit unterscheiden. In Abbildung 1 ist vereinfacht dargestellt, inwiefern sich zentrale ESBs von einer differenzierten Integrationsbereitstellung unterscheiden.

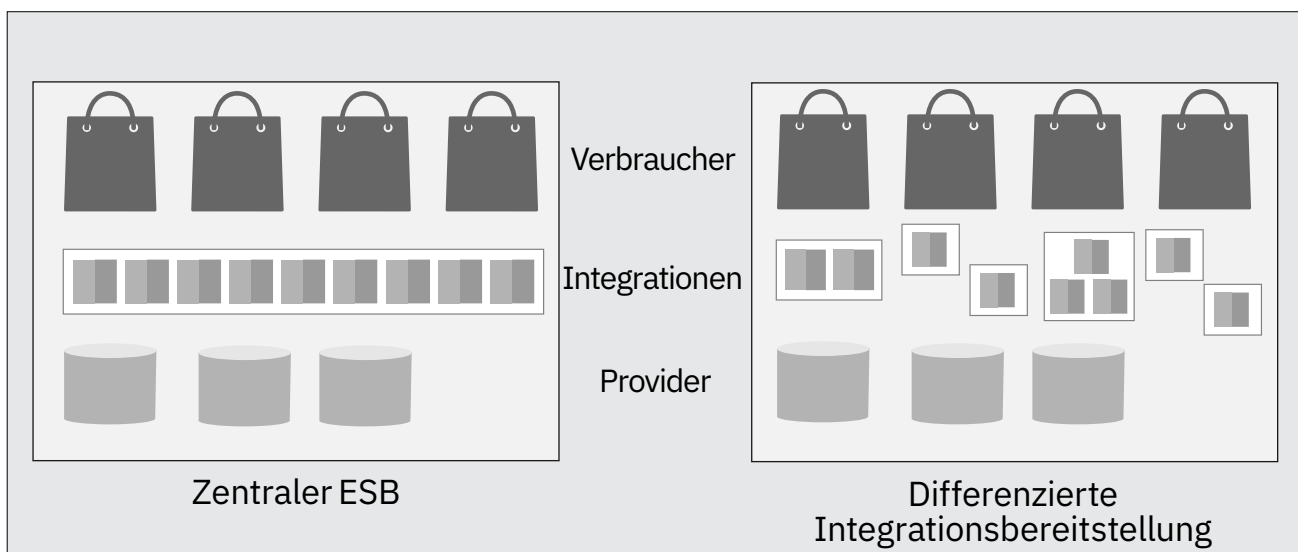


Abbildung 1: Vereinfachte Gegenüberstellung eines zentralen ESB und einer differenzierten Integrationsbereitstellung.

Eine differenzierte Integrationsbereitstellung macht sich die Vorteile einer Mikroservices-Architektur zunutze. Sehen wir uns die genannten Vorteile von Mikroservices noch einmal mit Blick auf die differenzierte Integrationsbereitstellung an:

- **Agilität** – Mehrere Teams können unabhängig voneinander und ohne Beteiligung einer zentralen Gruppe oder Infrastruktur, die zu einem Engpass werden könnte, an Integrationen arbeiten. Separate Integrationsströme können unabhängig von anderen Strömen geändert, neu erstellt und bereitgestellt werden. Dies führt zu mehr Sicherheit bei der Anwendung von Änderungen und beschleunigt den Produktionsstart.
- **Skalierbarkeit** – Separate Integrationsströme können einzeln skaliert werden, sodass Sie die effiziente elastische Skalierung von Cloudinfrastrukturen nutzen können.
- **Ausfallsicherheit** – Isolierte Integrationsströme, die in separaten Containern bereitgestellt werden, können sich nicht gegenseitig beeinträchtigen, indem sie sich gemeinsame Ressourcen wie Speicherkapazität, Verbindungen oder CPU-Leistung streitig machen.

Im Zusammenhang mit Agilität, Skalierbarkeit und Ausfallsicherheit darf nicht vergessen werden, dass diese Vorteile einer differenzierten Integration ohne dezentrale Integration nicht realisiert werden könnten.

[Viele weitere Informationen](#) über die differenzierte Integration finden Sie in unserem Buch „Agile Infrastructure Architecture“. Hier können Sie es [herunterladen!](#)

### Aspekt 2: Dezentrale Integrationsverantwortung

Eine große Herausforderung der serviceorientierten Architektur bestand darin, dass sie zur Erstellung der Serviceebene die Bildung zentraler Integrationsteams und Infrastrukturen erforderte.

Dies führte zu kontinuierlichen Problemen mit der Geschwindigkeit der Projektdurchführung, da ständig eine Abhängigkeit vom zentralen Integrationsteam bestand. Das zentrale Team kannte sich zwar gut mit der Integrationstechnologie aus, verstand jedoch häufig die zu integrierenden Anwendungen nicht. Dadurch kam es bei der Umsetzung der Anforderungen zu Verzögerungen und einer höheren Fehleranfälligkeit.

Viele Unternehmen hätten es vorgezogen, wenn die Anwendungsteams die Verantwortung für die Erstellung der eigenen Services getragen hätten, die Technologie und die Infrastruktur ließen das zu dieser Zeit jedoch nicht zu.

Mit dem Wechsel zur differenzierten Integrationsbereitstellung bietet sich die Möglichkeit, die Verantwortung für die Erstellung und Verwaltung von Integrationen zu verteilen. Es ist durchaus sinnvoll, dass Geschäftsanwendungsteams Integrationsaufgaben übernehmen und die Implementierung neuer Funktionen optimieren.

Haben wir Sie auf die differenzierte Integrationsbereitstellung neugierig gemacht? Unser Buch „[Agile Infrastructure Architecture](#)“ ist jetzt verfügbar und beantwortet Ihre Fragen!

### Aspekt 3: Cloudnative Integrationsinfrastruktur

Integrations-Laufzeitmodule haben sich in den vergangenen Jahren stark verändert. So stark, dass diese schlanken Laufzeitmodule sich in einem echt cloudnativen Kontext einsetzen lassen. Dabei beziehen wir uns auf ihre Fähigkeit, einen großen Teil der Belastung durch zuvor proprietäre Mechanismen für Cluster-Management, Skalierung und Verfügbarkeit an die Cloudplattform abzugeben, auf der sie ausgeführt werden.

Dies umfasst weit mehr als die Ausführung in einer containerisierten Umgebung. Es bedeutet, dass sie in der Lage sein müssen, als „cattle“ (gemäß der Metapher „cattle not pets“, deutsch: Vieh nicht Haustiere) zu funktionieren und Orchestrierungslösungen wie Kubernetes und andere gängige Standardframeworks für die Cloud optimal zu nutzen.

---

*Die Anwendung eines „cattle-Ansatzes“ beeinflusst die Art und Weise, wie Ihre DevOps-Teams mit der Umgebung und der Lösung insgesamt interagieren, und führt mit der Verlagerung weiterer Lösungen in schlanke Architekturen zunehmend zu Effizienzgewinnen.*

---

### Wie haben sich moderne Integrations-Laufzeitmodule mit Blick auf die agile Integrationsarchitektur verändert?

Eine agile Integrationsarchitektur setzt in jedem Fall voraus, dass die Integrationstopologie auf vollkommen andere Weise bereitgestellt wird. Ein wesentlicher Aspekt ist in diesem Zusammenhang ein modernes Integrations-Laufzeitmodul, das in einer containerbasierten Umgebung ausgeführt werden kann und sich gut für cloudnative Bereitstellungsverfahren eignet. Moderne Integrations-Laufzeitmodule haben mit ihren Gegenstücken aus früheren Zeiten fast nichts mehr gemein. Dies sind einige der Unterschiede:

- **Schnelle, schlanke Laufzeitmodule:** Sie werden in Containern, z. B. Docker, ausgeführt und lassen sich aufgrund ihrer Schlankheit in wenigen Sekunden starten und stoppen. Hinzu kommt, dass sie mithilfe von Orchestrierungs-Frameworks wie Kubernetes einfach zu verwalten sind.
- **Keine Abhängigkeit:** Sie brauchen keine Datenbanken oder Nachrichtenwarteschlangen mehr, sind jedoch offensichtlich sehr versiert darin, bei Bedarf entsprechende Verbindungen herzustellen.
- **Installation auf Dateisystembasis:** Zur Installation genügt es, die Binärdateien in einem Dateisystem abzulegen und sie zu starten. Dies ist optimal für die geschichteten Dateisysteme von Docker-Images.
- **Unterstützung für DevOps-Tools:** Das Laufzeitmodul muss eine kontinuierliche Integration ermöglichen und für die Bereitstellung vorbereitet sein. Installation, Erstellung, Bereitstellung und Konfiguration erfolgen mithilfe von Scripten und ‘Property’-Dateien, um „Infrastructure as Code“-Praktiken zu ermöglichen. Zur Beschleunigung der Aufnahme in DevOps-Pipelines sollten Vorlagenscripts für standardmäßige Build- und Bereitstellungstools zur Verfügung gestellt werden.
- **API-first:** Als primäres Kommunikationsprotokoll bieten sich RESTful APIs an. Die Bereitstellung von Integrationen als RESTful APIs dürfte problemlos möglich sein und auf gängigen Konventionen wie der Open API-Spezifikation basieren. Das Aufrufen von nachgeordneten RESTful APIs einschließlich der Erkennung mittels Definitionsdateien dürfte problemlos möglich sein.
- **Digitale Konnektivität:** Neben der reichhaltigen Unternehmenskonnektivität, die schon immer von Integrations-Laufzeitmodulen bereitgestellt wurde, muss auch eine Anbindung an moderne Ressourcen möglich sein. Dies gilt z. B. für NoSQL-Datenbanken (MongoDB und Cloudant usw.) und Nachrichtenservices wie Kafka. Darüber hinaus muss die Möglichkeit zum Zugriff auf einen umfangreichen Katalog von anwendungsintelligenten Konnektoren für SaaS-Anwendungen (Software as a Service), z. B. Salesforce, bestehen.

- **Continuous Delivery:** Continuous Delivery wird durch Befehlszeilenschnittstellen und 'Script templates' ermöglicht, die sich problemlos in standardmäßige DevOps-Pipeline-Tools einbinden lassen. Infolgedessen sind weniger spezialisierte Kenntnisse zur Implementierung von Schnittstellen erforderlich und die Bereitstellung wird beschleunigt.
- **Erweiterte Tools:** Erweiterte Integrationstools führen dazu, dass die meisten Schnittstellen allein durch Konfiguration erstellt werden können, und zwar auch von Personen ohne Integrationshintergrund. Durch das Hinzufügen von Vorlagen für gängige Integrationsmuster werden bewährte Integrationsverfahren fest in die Tools integriert, was die Aufgaben weiter vereinfacht. Integrationspezialisten werden weniger häufig benötigt. Unter Umständen könnte die Integration sogar teilweise von den Anwendungsteams übernommen werden, wie der folgende Abschnitt zur dezentralen Integration zeigt.

Moderne Integrations-Laufzeitmodule sind perfekt auf die drei Aspekte der agilen Integrationsarchitektur abgestimmt: differenzierte Bereitstellung, dezentrale Verantwortung und echt cloudnative Infrastruktur.

Sie möchten noch mehr über die cloudnative Infrastruktur erfahren? [Dann laden Sie jetzt unser Buch „Agile Integration Architecture“ herunter!](#)

## Agile Integrationsarchitektur für die Integrationsplattform

Bislang haben wir uns in diesem Whitepaper auf die Features der Anwendungsintegration konzentriert, die in einer agilen Integrationsarchitektur bereitgestellt werden. Viele Unternehmenslösungen erfordern jedoch die Anwendung mehrerer kritischer Integrationsfunktionalitäten. Eine Integrationsplattform (die von manchen Analysten auch als „hybride Integrationsplattform“ bezeichnet wird) vereint diese Funktionalitäten und ermöglicht Unternehmen damit größere Effizienz und Konsistenz bei der Erstellung von Geschäftslösungen.

Viele Branchenspezialisten sind vom Nutzen dieser Integrationsplattform überzeugt. Gartner bemerkt hierzu:

Die hybride Integrationsplattform (HIP) ist ein Framework aus On-Premises- und cloudbasierten Integrations- und Governance-Funktionen, das unterschiedlich qualifizierten Personen (Integrationspezialisten und Nichtspezialisten) die Unterstützung einer Vielzahl von Integrationsanwendungsfällen ermöglicht ... Die für die Integration verantwortlichen Anwendungsmanager sollten das HIP-Funktionsframework nutzen, um ihre Integrationsstrategien und -infrastrukturen zu modernisieren und sich damit für die neuen Anwendungsfälle im digitalen Unternehmen zu wappnen.<sup>2</sup>

Einer der wichtigsten Punkte, den Gartner nennt, ist der, dass die Integrationsplattform mehreren Mitarbeitern aus unterschiedlichen Bereichen des Unternehmens die Möglichkeit bietet, in einer ihren Anforderungen entsprechenden Benutzerumgebung zu arbeiten. Das heißt, dass Nutzern ohne tiefes IT-Wissen ein einfach zu bedienendes Integrations-Tooling zur Verfügung steht. Während IT-Spezialisten ein umfassenderes Funktionsspektrum für die Bearbeitung komplexerer Unternehmensszenarios verwenden können. Anschließend können diese Benutzer zusammenarbeiten und dabei die gemeinsamen Assets wiederverwenden, während die Gesamt-Governance erhalten bleibt.

Der Abdeckung neuer Anwendungsfälle im Rahmen der digitalen Transformation kommt die gleiche Bedeutung zu wie der Unterstützung der verschiedenen Benutzercommunitys. Im Hauptteil dieses Dokuments werden diese neuen Anwendungsfälle untersucht. Zunächst gehen wir jedoch auf die wichtigsten Funktionen ein, die Bestandteile der Integrationsplattform sein müssen.

## Die IBM Cloud Integration-Plattform

IBM Cloud Integration fasst die wichtigsten Integrationsfunktionalitäten in einer kohärenten, einfachen, schnellen und bewährten Plattform zusammen. Diese Plattform ermöglicht Ihnen die Erstellung von leistungsfähigen Integrationen und APIs in wenigen Minuten, sie zeichnet sich durch erstklassige Leistung und Skalierbarkeit aus und bietet hervorragende End-to-End-Funktionen mit auf Unternehmen abgestimmter Sicherheit.

In der IBM Cloud IntegrationsPlattform haben wir die sechs wichtigsten spezialisierten Integrationsfunktionen zusammengeführt, von denen sich jede durch besondere Leistungsfähigkeit auszeichnet. Es handelt sich um:

**API- Management:** Übernimmt die Bereitstellung und Verwaltung von Geschäftsservices in Form wiederverwendbarer APIs für bestimmte unternehmensinterne und -externe Entwicklercommunitys. Mit einer API-Strategie beschleunigen Unternehmen die effektive gemeinsame Nutzung ihrer spezifischen Daten- und Service-Assets als Grundlage für neue Anwendungen und Geschäftschancen.

**Sicherheitsgateway:** Erweiterte Konnektivität und Integration über die Unternehmensgrenzen hinaus mit für demilitarisierte Zonen (DMZ) geeigneten Funktionen

**Anwendungsintegration:** Verbindet On-Premises- und Cloud-Anwendungen und -Datenquellen zur Koordinierung des Austauschs von Geschäftsinformationen, damit Daten verfügbar sind, wann und wo immer sie benötigt werden.

**Nachrichtenübermittlung:** Stellt sicher, dass Echtzeitinformationen überall und jederzeit verfügbar sind. Dazu bietet die Funktion die zuverlässige und E2E-verschlüsselte Übermittlung von Nachrichten ohne Nachrichtenverlust, Duplizierung oder komplexe Wiederherstellung im Fall eines System- oder Netzproblems.

**Datenintegration:** Ermöglicht mit dem Zugriff, der Bereinigung und der Aufbereitung von Daten die Bereitstellung einer konsistenten Sicht auf Ihre Geschäftsabläufe in einem Data-Warehouse oder 'Data Lake' für die Durchführung von Analysen.

**Hochgeschwindigkeitsübertragung:** Ermöglicht die schnelle und vorhersehbare Übertragung riesiger Datenvolumen zwischen On-Premises- und Cloud-Standorten oder von Cloud zu Cloud mit hoher Sicherheitsstufe. Bestimmt die Geschwindigkeit, mit der Unternehmen Cloudplattformen einführen können, wenn sehr große Datenvolumen synchronisiert und ausgetauscht werden müssen.

Wir hoffen, dass Sie mit diesem Teaser-Whitepaper einen breiteren Überblick über die verschiedenen kritischen Funktionen gewonnen haben, die Bestandteil einer Integrationsplattform sein müssen. Und, dass Sie erkannt haben, wie Sie durch Einführung der agilen Integrationsarchitektur eine höhere Agilität, Skalierbarkeit und Ausfallsicherheit der Plattform erreichen können.

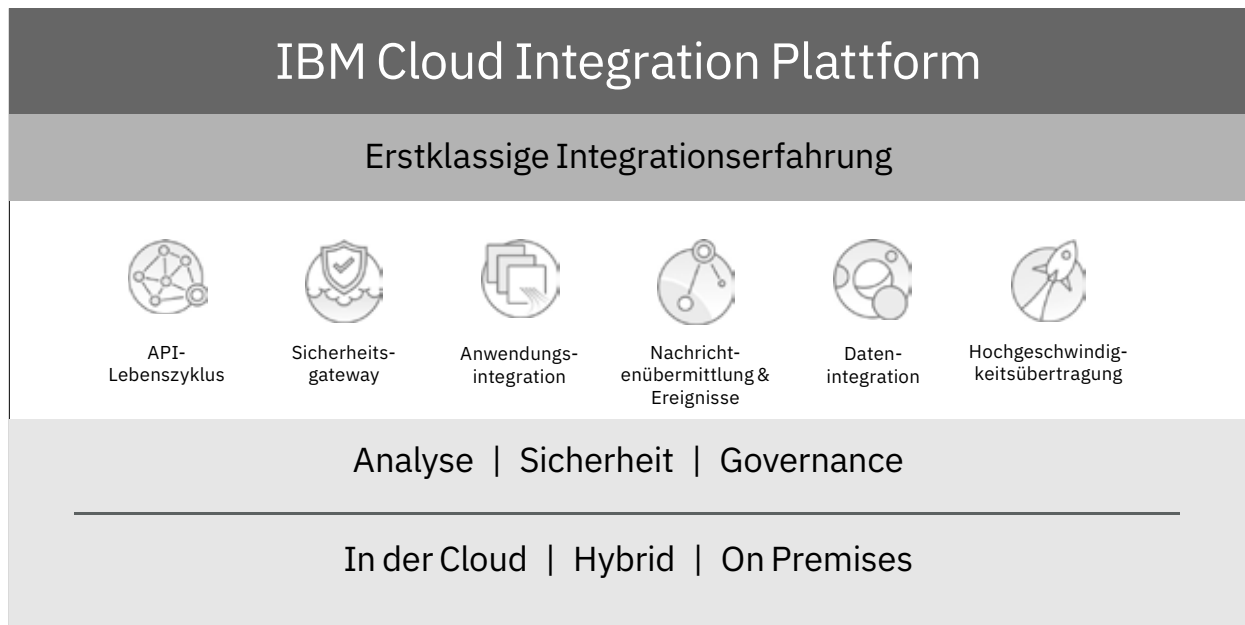


Abbildung 2: Die IBM Cloud Integration Plattform.

Laden Sie hier das umfassende [E-Book](#) herunter, um noch mehr über die agile Integrationsarchitektur zu erfahren.



---

**IBM Deutschland GmbH**

IBM-Allee 1  
71139 Ehningen  
[ibm.com/de](http://ibm.com/de)

**IBM Österreich**

Obere Donaustraße 95  
1020 Wien  
[ibm.com/at](http://ibm.com/at)

**IBM Schweiz**

Vulkanstrasse 106  
8010 Zürich  
[ibm.com/ch](http://ibm.com/ch)

Die IBM Homepage finden Sie unter:

**[ibm.com](http://ibm.com)**

IBM, das IBM Logo, [ibm.com](http://ibm.com), iSeries, Power, System Storage, zEnterprise, TDMF, AIX, BladeCenter und pSeries sind eingetragene Marken oder Marken der IBM Corporation in den USA und/oder anderen Ländern. Weitere Produkt- und Servicennamen können Marken von IBM oder anderen Unternehmen sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite „Copyright and trademark information“ unter [ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml)

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Microsoft, Windows und Windows NT sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

Die in diesem Dokument enthaltenen Informationen (einschließlich Angaben zu Währungen ODER Preisen, die nicht die jeweils geltenden Steuern enthalten) sind nur zum Datum der Erstveröffentlichung des Dokuments aktuell und können jederzeit ohne vorherige Ankündigung geändert werden. Nicht alle IBM Angebote sind in jedem Land, in welchem IBM tätig ist, verfügbar.

Leistungsdaten und Kundenbeispiele dienen nur zur Veranschaulichung. Die tatsächlichen Leistungsergebnisse können je nach Konfiguration und Betriebsbedingungen variieren.

Der Benutzer ist dafür verantwortlich, den Betrieb von Produkten oder Programmen anderer Anbieter in Verbindung mit IBM Produkten und Programmen zu prüfen und zu verifizieren.

Die Informationen in diesem Dokument werden auf der Grundlage des gegenwärtigen Zustands (auf „as-is“-Basis) ohne jegliche ausdrückliche oder stillschweigende Gewährleistung zur Verfügung gestellt, einschließlich, aber nicht beschränkt auf die Gewährleistungen für die Handelsüblichkeit, die Verwendungsfähigkeit für einen bestimmten Zweck oder die Freiheit von Rechten Dritter. Für IBM Produkte gelten die Gewährleistungen, die in den Vereinbarungen vorgesehen sind, unter denen sie erworben werden.

Die tatsächlich verfügbare Speicherkapazität kann sich sowohl auf nicht komprimierte als auch auf komprimierte Daten beziehen, sie kann Schwankungen unterliegen und geringer sein als angegeben.

© Copyright IBM Corporation 2018



Bitte der Wiederverwertung zuführen

---