



# Programmieren auf der SAP® BTP

- › Ihr Einstieg in die Entwicklung auf der SAP Business Technology Platform
- › Programmieren mit CAP, ABAP Cloud und SAP Build
- › Entwicklungsumgebungen, Programmiermodelle und KI-Integration

Pohl · Tate Tsiledze · Witte  
Teuffel von Birkensee



Rheinwerk  
Publishing

# Einleitung

Die *SAP Business Technology Platform* (SAP BTP) hat sich in den vergangenen Jahren zum zentralen Dreh- und Angelpunkt der modernen SAP-Welt entwickelt. Sie bildet das technologische Fundament für zahlreiche SAP-Lösungen und ermöglicht mit ihren vielfältigen Services – von der Anwendungsentwicklung über Integration, Prozessautomatisierung bis hin zur künstlichen Intelligenz – eine gewinnbringende Datennutzung und die effiziente Umsetzung innovativer Prozesse im Unternehmensumfeld.

Als offene und modulare Plattform bietet die SAP BTP entscheidende Vorteile, die Unternehmen dabei unterstützen, Innovationen und Prozesse schnell umzusetzen. Dazu zählt die *Clean-Core-Strategie*, bei der alle Anpassungen und Erweiterungen innerhalb der SAP-Landschaft auf bzw. mit der SAP BTP außerhalb des Kernsystems umgesetzt werden. Ebenso wichtig ist die *Multi-Cloud-Strategie*, die die Nutzung zusätzlicher Anbieter wie Amazon Web Services, Azure oder Google Cloud ermöglicht. Zudem ist die SAP BTP essenziell für die Einbindung von Zukunftstechnologien wie künstlicher Intelligenz, Automatisierung und event-getriebenen Architekturen.

## Ziel dieses Buches

Da die SAP BTP zahlreiche Services und damit eine Vielzahl an Technologien und Werkzeugen vereint, wirkt deren Zusammenspiel auf viele Anwender\*innen und Entwickler\*innen zunächst äußerst komplex.

Das Buch »Programmieren auf der SAP BTP« wurde geschrieben, um einen verständlichen und praxisnahen Leitfaden bereitzustellen, der Anwender\*innen und Entwickler\*innen bei der Entwicklung und Integration von Anwendungen auf der SAP BTP Schritt für Schritt begleitet und unterstützt. Dabei werden nicht nur die wichtigsten Grundlagen und Werkzeuge vorgestellt, sondern auch Best Practices und konkrete Handlungsanweisungen aufgezeigt – verständlich, anwendbar und direkt in realen Projekten umsetzbar.

## Für wen ist dieses Buch geeignet?

Das Werk richtet sich in erster Linie an Anwender\*innen und Entwickler\*innen, die auf der SAP BTP programmieren oder Erweiterungen umsetzen möchten, sowie an Berater\*innen und Entscheider\*innen, die ein grundlegendes Verständnis der SAP BTP und ihrer vielfältigen Einsatzmöglichkeiten erlangen möchten. Auch Studie-

rende und Berufseinsteiger\*innen im SAP-Umfeld finden in diesem Werk einen fundierten Überblick und eine praktische Anleitung.

## Inhalt dieses Buches

**Kapitel 1**, »SAP Business Technology Platform: strategische Grundlagen und Architektur«, führt zunächst in den Clean-Core-Ansatz ein, eines der zentralen Leitprinzipien der SAP BTP. Es wird detailliert erläutert, wie mit Key-User-Erweiterungen, ABAP-Cloud- und Side-by-Side-Erweiterungen standardisierte und zukunftsfähige Architekturen gestaltet werden können.

**Kapitel 2**, »SAP Cloud Application Programming Model«, erläutert das Thema SAP Cloud Application Programming Model (CAP) und zeigt, wie Sie mit diesem Programmiermodell leistungsstarke Anwendungen entwickeln können. Außerdem wird erläutert, wie damit Datenmodelle und Services strukturiert sowie SAP-Fiori- und andere Frontends nahtlos integriert werden.

**Kapitel 3**, »ABAP in der Cloud und ABAP RESTful Application Programming Model«, widmet sich dem Schwerpunkt ABAP in der Cloud. Es wird herausgearbeitet, wie sich die ABAP-Umgebung von klassischen On-Premise-Systemen unterscheidet und wie mit dem ABAP RESTful Application Programming Model zukunftsfähige und skalierbare Services und SAP-Fiori-Anwendungen entstehen.

SAP Build steht im Zentrum von **Kapitel 4**, »SAP Build«. Die SAP-Build-Lösungen – SAP Build Apps, SAP Build Process Automation, SAP Build Work Zone und SAP Build Code – ermöglichen eine nahtlose Verbindung von No-Code-, Low-Code- und Pro-Code-Entwicklungen. Dieses Zusammenspiel bildet eine perfekte Grundlage für die Zusammenarbeit zwischen SAP-Anwender\*innen und professionellen Entwickler\*innen.

**Kapitel 5**, »Events und APIs«, zeigt die Bedeutung event-getriebener Architekturen auf. Anhand des Beispiels SAP Event Mesh wird erläutert, wie sich verschiedene Systeme effizient vernetzen lassen und welche Chancen sich dadurch eröffnen.

In **Kapitel 6**, »Künstliche Intelligenz auf der SAP BTP«, werden die Einsatzmöglichkeiten von künstlicher Intelligenz auf der SAP BTP erläutert. Es werden diverse KI-Services vorgestellt, wie z. B. SAP Document AI, Data Attribute Recommendation und generative Modelle. Darüber hinaus wird gezeigt, wie mit SAP AI Core, SAP AI Launchpad und Joule eigene Szenarien umgesetzt und native Machine-Learning-Funktionen von SAP HANA Cloud genutzt werden können.

Zuletzt wird im abschließenden **Kapitel 7**, »DevOps und Sicherheit«, das Thema DevOps auf der SAP BTP näher beleuchtet. Es wird gezeigt, wie der Einsatz von Git, CI/CD

Pipelines und SAP-Transportmanagement moderne Entwicklungs- und Betriebsprozesse etablieren kann. Ergänzend werden Tools wie der SAP Alert Notification Service und Kibana vorgestellt.

### Programmierbeispiele zum Download

Damit Sie die Beispiele in diesem Buch nachvollziehen und sofort selbst ausprobieren können, finden Sie unter [www.sap-press.de/6104](http://www.sap-press.de/6104) im Bereich **Materialien** die Programmierbeispiele zum Download.



In hervorgehobenen *Informationskästen* finden Sie in diesem Buch Inhalte, die wissenswert und hilfreich sind, aber etwas außerhalb der eigentlichen Erläuterung stehen. Damit Sie die Informationen in den Kästen sofort einordnen können, haben wir die Kästen mit Symbolen gekennzeichnet:

Dieses Symbol weist Sie auf *Besonderheiten* hin, die Sie beachten sollten. Es *warnt* Sie außerdem vor häufig gemachten Fehlern oder Problemen, die auftreten können.



Mit diesem Symbol sind *Tipps* und *Hinweise* aus der Berufspraxis markiert, die praktische Empfehlungen geben, die Ihnen die Arbeit erleichtern können.



In Kästen, die mit diesem Symbol gekennzeichnet sind, finden Sie Informationen zu *weiterführenden Themen* oder wichtigen Inhalten, die Sie sich merken sollten.



## Danksagung

Ein Fachbuch wie dieses, geschrieben von Experten, die nicht hauptberufliche Autoren, sondern vor allem auch Kollegen, Vorgesetzte, Partner und Väter sind, entsteht nicht im Vakuum. Unzählige Stunden konzentrierter Schreibarbeit flossen in die sieben Kapitel dieses Werks. »Programmieren auf der SAP BTP« wäre nicht möglich gewesen ohne die Unterstützung zahlreicher Menschen in unserem persönlichen Umfeld, denen wir im Folgenden in persönlichen Worten von Herzen danken möchten.

### Jean-François Tate Tsiledze:

Zunächst danke ich meiner Mutter Martine von Herzen für ihre liebevolle Erziehung und all die Opfer, die sie für mich gebracht hat. Ein besonderer Dank gilt meiner Frau Nadja für ihre unermüdliche Unterstützung und dafür, dass sie mir so vieles abgenommen hat, um mir Raum für dieses Buch zu geben. Meine Söhne Christophe und Raphaël – ihr seid meine größte Motivation und gebt mir die Kraft, auch an schweren Tagen weiterzumachen.

**Florian Pohl:**

Mein Dank gilt ganz besonders meiner Verlobten Alina für ihre starke Unterstützung, ihr Verständnis und die Kraft, die sie mir während der Entstehung dieses Buches gab. Ich danke meiner Familie und meinen Freunden für ihr Verständnis während dieser intensiven Zeit und dafür, dass sie mir trotz meiner begrenzten Verfügbarkeit wertvolle Ablenkung verschafften. Darüber hinaus danke ich auch meinem Teamleiter Martin Pichlo herzlichst für die nötige Zeit, die er mir in den vergangenen Wochen zum Schreiben einräumte.

**Raphael Witte:**

Mein besonderer Dank gilt meiner Frau Daniela und unserer Tochter Sophia. Während der Arbeit an diesem Buch blieb oft wenig gemeinsame Zeit. Umso mehr möchte ich mich bei euch für eure Unterstützung in dieser Zeit bedanken. Ich möchte mich auch herzlich bei meiner Mutter Hedwig und meinem verstorbenen Vater Johannes bedanken – ohne euch wäre ich heute nicht da, wo ich bin. Dank gilt auch meiner Familie und meinen Freunden für Ihr Verständnis und die oft willkommene Ablenkung. Für die Unterstützung und die vielen hilfreichen Gespräche bedanke ich mich zudem herzlichst bei meinen Kolleg\*innen im Cloud Custom Development.

**Constantin Freiherr Teuffel von Birkensee:**

Ein besonderer Dank gilt meinen beiden Kindern, Amelie und Maxime, die mich in dieser intensiven Zeit nicht nur begleitet, sondern auch mit viel Geduld und Verständnis unterstützt haben. Eure Liebe und euer Rückhalt haben mir viel Kraft gegeben. Ebenso möchte ich mich herzlich bei meiner Ex-Frau Viktoria bedanken. Sie hat mir in vielerlei Hinsicht den Rücken frei gehalten und mich, gemeinsam mit ihrem Mann, tatkräftig unterstützt – dafür bin ich sehr dankbar. Ein großes und von Herzen kommendes Dankeschön geht auch an meine Eltern, Birgit und Benedikt Teuffel von Birkensee. Eure Liebe, euer Vertrauen und eure stetige Unterstützung haben mir den Weg geebnet. Ohne euch wäre ich heute nicht da, wo ich bin.

**Gemeinsamer Dank:**

Ein herzliches Dankeschön geht an unsere Vorgesetzten, Markus Böhme und Eric Rotzoll, die uns nicht nur die Freiheit gegeben haben, dieses Buch zu schreiben, sondern uns auch mit Vertrauen, Rückhalt und wertvollem Feedback unterstützt haben. Ohne diese Offenheit und Förderung wäre »Programmieren auf der SAP BTP« in die-

ser Form nicht möglich gewesen. Wir bedanken uns zudem bei Tsvetelina Tsoneva und Janine Baumann – eure Unterstützung, euer offenes Ohr und eure Bereitschaft, uns in vielen kleinen und großen Dingen zu helfen, haben uns sehr weitergebracht und motiviert.

Zum Schluss danken wir der gesamten SAP-Community. Die zahlreichen Beiträge, Diskussionen und das geteilte Wissen von Expert\*innen, Anwender\*innen und Vor- denker\*innen waren für uns eine stetige Quelle der Inspiration und haben maßgeblich zur Qualität und Tiefe dieses Buches beigetragen.

# Kapitel 1

## SAP Business Technology Platform: strategische Grundlagen und Architektur

*Die SAP Business Technology Platform eröffnet Ihnen viele Möglichkeiten. Das Konzept des Clean Core bildet das neue Fundament der Entwicklung. In diesem Kapitel erfahren Sie, wie Sie diesen Ansatz für Ihre spezifischen Szenarien optimal nutzen können.*

Standards vereinfachen Geschäftsprozesse erheblich. SAP bietet für die meisten Business-Szenarien standardisierte Szenarien, die Kunden direkt nutzen können. Viele Unternehmen verfügen jedoch über individuelle Prozesse, die sie gerade in ihrem Business auszeichnen und besonders machen. Diese Prozesse machen den Unterschied – doch wie lassen sich solche kundenindividuellen Prozesse im SAP-System umsetzen?

Im klassischen SAP ECC (R3) konnten Systeme umfassend modifiziert werden, was allerdings häufig Updates und Upgrades erschwerte oder unmöglich machte. In der modernen cloud-getriebenen Systemlandschaft ist dieser Ansatz weder möglich noch empfehlenswert. Der Grund: System-Updates und neue Releases sind heute wichtiger denn je, neue Technologien wie künstliche Intelligenz oder Automatisierungen erfordern regelmäßige Aktualisierungen, um wettbewerbsfähig zu bleiben. Bestimmte Anpassungen oder gar Modifikationen sind in Cloud-Systemen heutzutage nicht mehr vorgesehen. Dieses Kapitel zeigt, wie Sie mit der SAP Business Technology Platform (SAP BTP) dennoch eigene Szenarien umsetzen können, ohne Flexibilität einzubüßen. Dazu stellen wir Ihnen die folgenden Themen vor:

- Grundprinzipien und Möglichkeiten von Clean Core
- Einblick in die verschiedenen Entwicklungsmöglichkeiten und -plattformen
- Vorstellung der SAP Business Technology Platform
- Überblick von Events, API und AI

Abbildung 1.1 veranschaulicht die typischen Bedürfnisse bei Kunden.

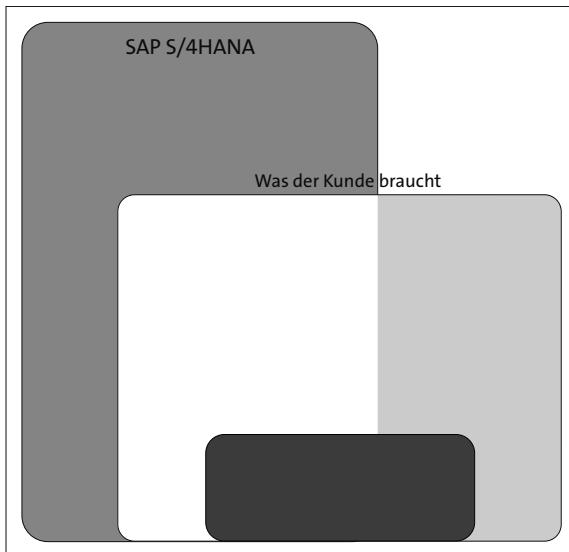


Abbildung 1.1 Entwicklungsanforderung beim Kunden

Kernziel ist die klare Trennung zwischen SAP-Entwicklung und Kundenentwicklungen, um Eingriffe in den SAP-Programmablauf zu vermeiden. Diese Trennung reduziert den Upgrade-Aufwand erheblich und ermöglicht automatische Updates in Cloud-Systemen. Dabei kommen moderne Technologien zum Einsatz:

- SAP Cloud Application Programming Model (CAP)
- ABAP RESTful Application Programming Model
- In-App Extensions

Abschnitt 1.1 erläutert die Clean-Core-Strategie und zeigt die kritischen Erfolgsfaktoren für deren Implementierung.

## 1.1 Clean Core

Das Ziel von Clean Core ist, dass das System so nah wie möglich im Standard verbleibt und die Erweiterungen cloud-kompatibel und integriert sind. Dabei ist es erlaubt, das System zu adaptieren, um die Geschäftsanforderung umzusetzen und neue Funktionen zu erweitern. Clean Core steht für ein Systemarchitekturkonzept und eine Philosophie, die sich in verschiedenen Dimensionen streng am Standard orientiert.

Zu den Zielen von Clean Core gehören:

- Schaffung einer reaktionsfähigen IT-Architektur zur flexiblen Anpassung an sich ändere Geschäftsanforderungen

- Ermöglichen problemloser System-Upgrades
- Bereitstellung einer Grundlage für cloud-konforme, modulare Erweiterungen und Integrationen
- Vorbereitung für einen potenziellen zukünftigen Wechsel in die Public Cloud

### 1.1.1 Grundprinzipien des Clean-Core-Ansatzes

Clean Core geht über »saubere« Programmierung hinaus und umfasst die gesamte Systemarchitektur sowie die dazugehörigen Prozesse.

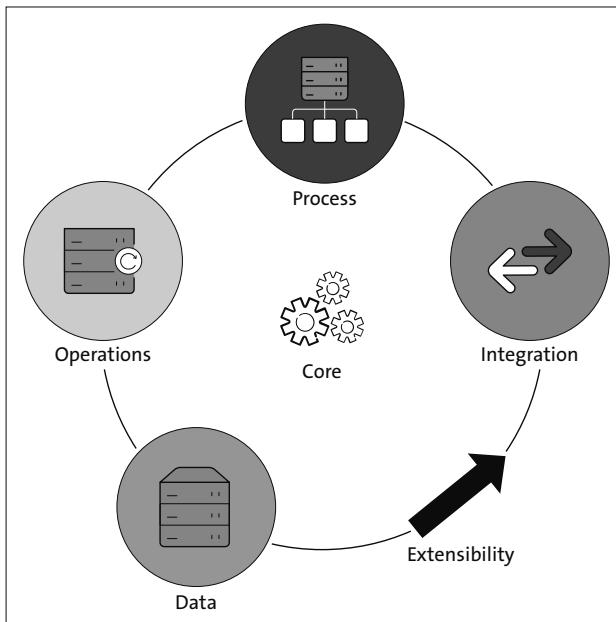


Abbildung 1.2 Clean Core

SAP definiert fünf Bereiche von Clean Core (siehe Abbildung 1.2):

1. **Extensibility:** effektive, upgrade-stabile Erweiterungen, die von SAP-Standardfunktionen entkoppelt sind
2. **Data:** hohe Datenqualität und Governance zur Ausschöpfung des Prozesspotenzials und Minimierung von Fehlern
3. **Integrations:** erweiterbare IT-Landschaft, die innovative APIs sowie ein ereignisgesteuertes Design nutzt
4. **Processes:** Optimierung der Prozesse, um die Wettbewerbsfähigkeit zu erhöhen und die Komplexität zu minimieren
5. **Operations:** Implementierung von bewährten Betriebskonzepten, Systemüberwachung und geeignetem Release-Management

Diese fünf Dimensionen haben unterschiedliche Gewichtungen im Bereich Entwicklung und sind für alle Beteiligten in SAP-Projekten relevant.

### Extensibility

Erweiterungen sollten nur erfolgen, wenn es nötig ist. Sie müssen cloud-kompatibel sein und mithilfe des Three-Tier-Modells (siehe Abschnitt 1.1.3) entwickelt werden. Es muss sichergestellt werden, dass immer der volle Umfang der Erweiterungsmöglichkeiten (siehe Abschnitt 1.1.2, »Technologische Grundlagen: In-App Extensions, ABAP Cloud und Side-by-Side Extensions«) genutzt wird und man sich nicht auf eine Technologie beschränkt. Für Erweiterungen müssen klare Anforderungen und Richtlinien vorhanden sein, wie die Entwicklung durchgeführt werden soll. Technische Unsicherheiten müssen dokumentiert und regelmäßig auf ihre Machbarkeit hin überprüft werden.

### Data

Um eine optimale Performance des Systems sicherzustellen, ist es notwendig, das Datenvolumen kontinuierlich zu kontrollieren und gegebenenfalls zu reduzieren. Dabei sollten veraltete, nicht mehr genutzte oder redundante Daten konsequent identifiziert und entfernt werden, um unnötige Belastungen für das System zu vermeiden. Besonders bei der Speicherung personenbezogener Daten muss sorgfältig geprüft werden, ob diese tatsächlich erforderlich ist. Es muss sichergestellt sein, dass solche Daten bei Bedarf gelöscht werden können, um sowohl gesetzlichen Anforderungen als auch unternehmensinternen Datenschutzrichtlinien gerecht zu werden.

### Integrations

Die Nutzung von Standard-APIs mit OData oder SOAP ist entscheidend für eine zukunftssichere und wartbare Systemarchitektur. Die Verwendung von veralteten Techniken wie RFC oder IDoc sollte vermieden werden, da diese nicht mehr aktuellen Anforderungen im Hinblick auf die Erweiterbarkeit und Cloud-Kompatibilität entsprechen. Stattdessen empfiehlt sich die Verwendung moderner, ereignisgesteuerter Designs für Standard-Events zur Integration.

### Processes

Die Definition von Prozessen ist ein entscheidender Faktor für die Clean-Core-Umsetzung. Berater\*innen und Kunden müssen sich intensiv mit den Clean-Core-Dimensionen auseinandersetzen, um sicherzustellen, dass Prozesse aktuellen Standards entsprechen und adäquat implementiert werden können.

## Operations

Clean Core wird vollständig im gesamten End-to-End-Prozess angewandt, um wartbare Systemlandschaften zu gewährleisten. Ein effektives Release Management spielt dabei eine zentrale Rolle, damit Änderungen geprüft und nachvollziehbar sind. Definierte Entwicklungsrichtlinien müssen zur Qualitätssicherung regelmäßig geprüft werden. Zusätzlich sollten Tools wie das ABAP Test Cockpit (ATC, siehe Abschnitt 1.1.5, »Tools und Services«) genutzt werden, um die Qualität der Entwicklung zu bestimmen und potenzielle Schwachstellen frühzeitig zu finden. Identifizierte Schwachstellen sollten dokumentiert und regelmäßig daraufhin überprüft werden, ob sie beseitigt werden können.

### 1.1.2 Technologische Grundlagen: In-App Extensions, ABAP Cloud und Side-by-Side Extensions

SAP bietet Entwickler\*innen verschiedene Möglichkeiten, mit Clean Core zu entwickeln:

- Key-User Extension oder auch In-App Extension genannt
- ABAP Cloud
- Side-by-Side Extensions

Abbildung 1.3 zeigt einen Gesamtüberblick aller Technologien, die für Clean Core bereitgestellt werden.

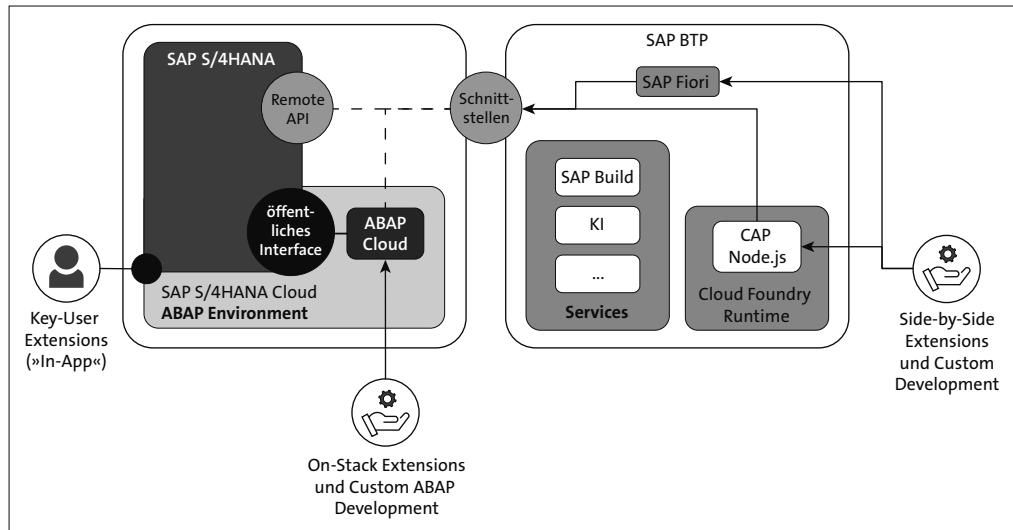


Abbildung 1.3 Technologische Grundlagen für Clean Core

Im Folgenden werden alle Technologien einzeln betrachtet, angefangen mit der Key-User Extension, die die geringsten Erweiterungsmöglichkeiten bietet, über ABAP Cloud bis zur Side-by-Side-Entwicklung, die die vielfältigsten Erweiterungsmöglichkeiten etabliert.

### Key-User Extensions

Key-User Extensions sind der einfachste und schnellste Weg in der Clean-Core-Strategie von SAP, um Kundenanforderungen in SAP S/4HANA umzusetzen, ohne den Standard zu beeinträchtigen. Die folgenden Tools ermöglichen es Entwickler\*innen und Key-Usern, Erweiterungen direkt im System vorzunehmen:

- **UI Adoption Mode:** Ermöglicht unkomplizierte Anpassungen der Benutzeroberfläche.
- SAP-Fiori-App **Custom Fields and Logic:** Erlaubt die Definition und Integration eigener Felder in Anwendungen.
- SAP-Fiori-App **Custom Business Objects:** Ermöglicht die Modellierung neuer Geschäftsmodelle für komplexere Anforderungen.
- SAP-Fiori-App **Custom CDS Views:** Unterstützt die Erstellung individueller Datenmodelle.
- SAP-Fiori-App **Custom Analytical Queries:** Dient der Erstellung analytischer Auswertungen.

### ABAP Cloud

Erweiterungen mit ABAP Cloud sind eine moderne Möglichkeit, SAP S/4HANA gezielt zu erweitern, ohne den Systemkern zu beeinträchtigen. Die Erweiterung erfolgt dabei innerhalb eines ABAP-Systems, das über freigegebene öffentliche Schnittstellen mit dem SAP-S/4HANA-System kommuniziert. Dieses Vorgehen ermöglicht es, individuelle Anforderungen umzusetzen, während gleichzeitig die Upgrade-Fähigkeit und Wartbarkeit des Systems erhalten bleiben. Voraussetzung für die Nutzung dieser Erweiterungsform ist eine dreistufige Systemlandschaft in SAP S/4HANA Public Cloud oder ein Release ab 2022. Die Zielgruppe sind ABAP-Entwickler\*innen, die mit den spezifischen Anforderungen und Möglichkeiten von ABAP Cloud vertraut sind. Der Anwendungsbereich umfasst insbesondere die Erstellung und Erweiterung von Anwendungen, die eng mit dem SAP-S/4HANA-System verbunden sind und eine hohe Systemauslastung aufweisen. ABAP Cloud kommt der klassischen ABAP-Entwicklung von den drei Möglichkeiten für Clean Core am nächsten. Dabei ist zu beachten, dass es sich nicht um klassisches ABAP handelt, sondern um ABAP Cloud (siehe dazu Kapitel 3, »ABAP in der Cloud und ABAP RESTful Application Programming Model«) und daher gewissen Neuerungen unterliegt.

Abbildung 1.4 verdeutlicht noch einmal, dass es sich um eine neue Form der On-Stack-Erweiterungen handelt, die nicht mehr direkt im SAP-S/4HANA-Kern, sondern daneben entwickelt wird.

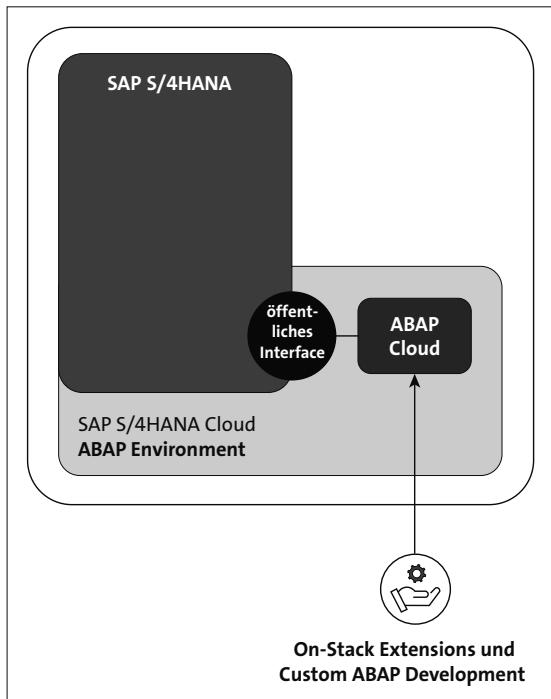


Abbildung 1.4 ABAP Cloud

### Side-by-Side Extensions

Side-by-Side Extensions sind die dritte und umfangreichste Möglichkeit der Clean-Core-Strategie und ermöglichen die Entwicklung individueller Funktionen außerhalb des SAP-S/4HANA-Kerns. Anders als die ersten beiden Möglichkeiten lassen sich Side-by-Side Extensions nicht nur im SAP-S/4HANA-Umfeld einsetzen, sondern können auch andere Cloud-Produkte erweitern. Im Gegensatz zu On-Stack-Erweiterungen, bei denen kundenspezifische Logik direkt im ERP-System implementiert wird, erfolgt die Entwicklung bei Side-by-Side-Ansätzen auf der SAP BTP. Diese Trennung sorgt für eine klare Abgrenzung zwischen Standard und kundenspezifischer Logik und erhöht die Upgrade-Fähigkeit sowie die langfristige Wartbarkeit der Systemlandschaft. Die Kommunikation zwischen SAP S/4HANA und den Erweiterungen auf der SAP BTP erfolgt über Remote APIs. Zusätzliche Schnittstellen können über ABAP Cloud oder auch über Key-User Extensions bereitgestellt werden. Dies gewährleistet eine kontrollierte und sichere Interaktion mit dem SAP-S/4HANA-System und schafft eine flexible Grund-

lage für die Integration externer Anwendungen und Services. Innerhalb der SAP BTP stehen moderne Entwicklungsframeworks wie das SAP Cloud Application Programming Model (CAP) mit Node.js oder Java zur Verfügung, über das Sie in Kapitel 2 mehr erfahren. Anwendungen können in der Cloud Foundry Runtime oder anderen Laufzeitumgebungen betrieben und durch Komponenten wie SAP Build, SAP AI Services und weitere SAP-BTP-Services ergänzt werden. Entwickler\*innen profitieren von einer cloud-basierten Umgebung, die sich nahtlos in bestehende DevOps-Prozesse integrieren lässt und eine schnelle Bereitstellung sowie Skalierung von Erweiterungen ermöglicht.

Abbildung 1.5 gibt Ihnen einen Überblick über die Side-by-Side-Entwicklung. Zusammenfassend ermöglichen Side-by-Side Extensions eine flexible, skalierbare und upgrade-sichere Entwicklung außerhalb des SAP-S/4HANA-Kerns. Sie sind ideal für komplexe Anforderungen, die moderne Cloud-Technologien und eine klare Trennung vom Standard erfordern.

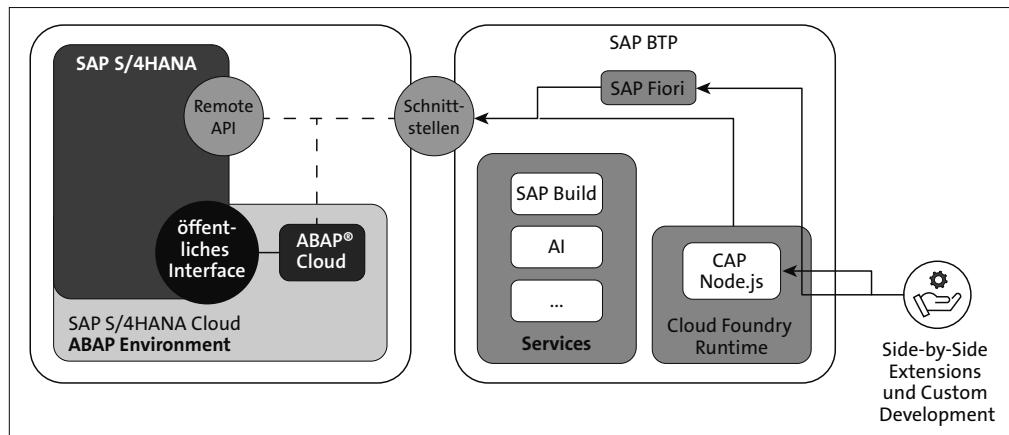


Abbildung 1.5 Side-by-Side Extensions

### 1.1.3 Das Three-Tier-Modell in der Praxis

Das *Three-Tier-Modell* beschreibt ein einheitliches Vorgehen zur Erweiterung von SAP-S/4HANA-Systemen unter Berücksichtigung von Clean Core. Es unterscheidet sich in der Lauffähigkeit der unterschiedlichen Systeme. Abbildung 1.6 gibt einen Überblick.

Das klassische ABAP entspricht Tier 3 – also der dritten Ebene –, da es Clean Core nicht unterstützt und auch nur als On-Premise-Version oder in der Private Cloud entwickelt werden kann. Auf den Ebenen 1 und 2 – Tier 1 und Tier 2 – kann auf jeglicher Plattform entwickelt werden, und es wird mit ABAP Cloud implementiert. Entwick-

lungen auf der SAP BTP sind automatisch durch ihre Side-by-Side-Entwicklung Tier 1 zugeordnet und gewährleisten somit vollständige Clean-Core-Konformität.

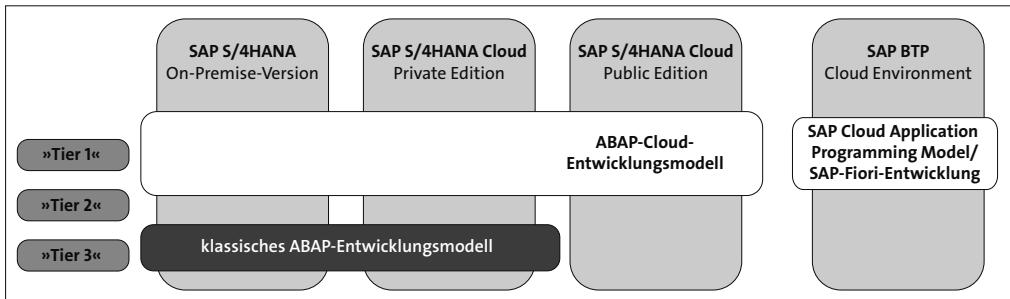


Abbildung 1.6 Three-Tier-Modell

### Tier 1

In der ersten Ebene – Tier 1 – dürfen ausschließlich eigene Objekte und von SAP offiziell freigegebene Objekte verwendet werden. Sowohl ABAP Cloud als auch Side-by-Side-Entwicklungen erfüllen diese Bedingungen und repräsentieren die ideale Umsetzung von Clean Core in der Entwicklung. Diese Konformität resultiert in maximaler Wartbarkeit und garantiert reibungslose System-Updates und -Upgrades ohne Kompatibilitätsprobleme.

### Tier 2

Die zweite Ebene – Tier 2 – folgt grundsätzlich denselben Prinzipien wie Tier 1, erlaubt jedoch in Ausnahmefällen den Zugriff auf nicht freigegebene SAP-Objekte. Dieser Zugriff muss jedoch über eine sogenannte *Wrapper-Klasse* erfolgen. Entwickelte Wrapper-Klassen dürfen ausschließlich von einer Funktion aufgerufen werden, die wiederum das nicht freigegebene Objekt referenziert. Der Einsatz von Wrapper-Klassen schafft eine Kapselung und reduziert eine direkte Abhängigkeit vom Kern.

### Tier 3

Die dritte Ebene – Tier 3 – repräsentiert den klassischen ABAP-Entwicklungsansatz mit uneingeschränktem Zugriff auf sämtliche Systemobjekte – unabhängig von deren offiziellem Freigabestatus. Während dieser Ansatz zwar eine größtmögliche Flexibilität für die Entwickler\*innen bietet, birgt er auch hohe Risiken hinsichtlich der Wartbarkeit sowie der Update- und Upgrade-Fähigkeit des Systems. Daher sollte diese Ebene nur in absoluten Ausnahmefällen noch eingesetzt werden.

Abbildung 1.7 zeigt den Zugriffsfluss von ABAP-Entwicklungen im Three-Tier-Modell.

Beispiele für die Nutzung des Three-Tier-Modells sind folgende:

- Tier 1
  - Hinzufügen von Custom Fields über Apps
  - Erweiterung oder Entwicklung einer SAP-Fiori-App unter Verwendung des ABAP RESTful Application Programming Models und von CAP
  - Erweiterung mit einem freigegebenen Business Add-in (BAdI)
- Tier 2
  - Erstellung von Wrapper-Klassen für nicht freigegebene Objekte
- Tier 3
  - Hinzufügen eines Custom Fields über Append
  - Erweiterung oder Entwicklung einer SAP-Fiori-App mit UI5, Service Gateway oder Business Object Processing Framework
  - Erweiterung mit einem nicht freigegebenen BAdI
  - Entwicklung von Reports, Dynpros oder Web Dynpros
  - Modifikationen durch User Exits oder Änderung des Quellcodes

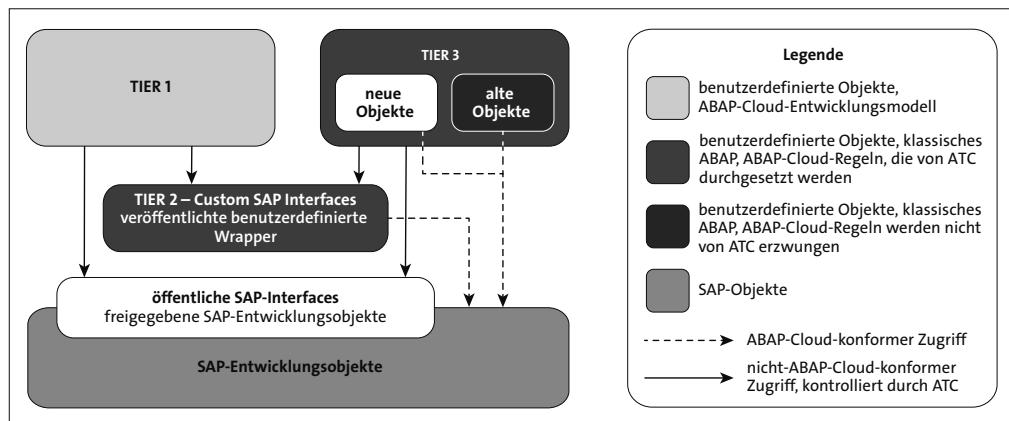


Abbildung 1.7 Beispiel von ABAP im Three-Tier-Modell

#### 1.1.4 Clean Core Quality Gates

Zur Validierung einer erfolgreichen Clean-Core-Implementierung hat SAP sogenannte *Clean Core Quality Gates* entworfen. Diese Gates müssen erfüllt werden, um den Status Clean Core zu erreichen. Die Quality Gates werden über die SAP-Roadmap mit begleitenden Tasks in SAP Cloud Application Lifecycle Management (SAP Cloud ALM) eingespielt. Abbildung 1.8 visualisiert die strukturelle Gliederung der Clean Core Quality Gates, wobei SAP insgesamt mehr als 300 spezifische Qualitätskriterien definiert hat.



Abbildung 1.8 Dimensionen der Quality Gates

### Zusammenarbeit mit Quality Gates

Clean Core und die Quality Gates sind nicht nur reine Entwicklungsthemen. Es empfiehlt sich, zu den vorhandenen Dimensionen weitere für die Bearbeitung der Gates hinzuzufügen, z. B. cross-funktionale Teams. Das erleichtert das Zusammenspiel zwischen Unternehmen, Entwickler\*innen und Berater\*innen, da es immer übergreifende Themen gibt, die für mehrere relevant sind.



Die Clean Core Quality Gates sind integraler Bestandteil aller sechs Phasen des SAP-Implementierungszyklus:

1. **Discover:** Verständnis für die SAP-Lösungen entwickeln und den Business Case definieren
2. **Prepare:** Projektteam zusammenstellen, initiale Planung und Systemzugänge vorbereiten
3. **Explore:** Fit-to-Standard-Workshops durchführen, um Geschäftsprozesse mit dem SAP-Standard abzulegen
4. **Realize:** Konfiguration, Erweiterungen und Tests auf Basis der definierten Anforderungen durchführen
5. **Deploy:** Produktivsetzung vorbereiten, Schulungen durchführen und Go-live realisieren
6. **Run:** den laufenden Betrieb sicherstellen und kontinuierliche Verbesserungen umsetzen

Die phasenspezifischen Clean Core Quality Gates müssen jeweils entsprechend dem aktuellen Projektfortschritt erfüllt werden. Daher sind die Clean Core Quality Gates ein kontinuierlicher Begleiter während des gesamten Projekts und können nicht erst am Ende betrachtet werden.

### 1.1.5 Tools und Services

Es gibt verschiedene Tools und Services, die Sie bei der Erreichung von Clean Core unterstützen. Die drei wichtigsten sind folgende:

- ABAP Test Cockpit (ATC)
- SAP Cloud ALM
- Entwicklungsrichtlinien

Während die Entwicklungsrichtlinien individuell durch Namenskonventionen etc. abgebildet und definiert werden müssen, bietet SAP mit dem ABAP Test Cockpit und SAP Cloud ALM standardisierte Lösungen zur systematischen Unterstützung der Clean-Core-Implementierung.

#### ABAP Test Cockpit (ATC)

SAP stellt im ABAP Test Cockpit zwei spezialisierte Prüfvarianten bereit, die gezielt auf die Erreichung von Clean-Core-Konformität ausgerichtet sind:

- ABAP\_CLOUD\_READINESS
- ABAP\_CLOUD\_DEVLEOPMENT\_DEFAULT

Zur Validierung der Cloud-Fähigkeit von ABAP-Entwicklungen wird die Prüfvariante »ABAP\_CLOUD\_READINESS« eingesetzt (siehe Abbildung 1.9).

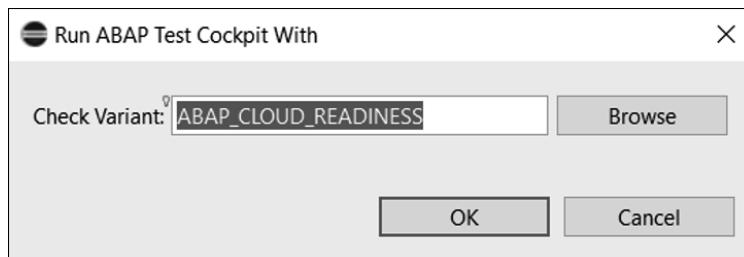


Abbildung 1.9 ABAP Test Cockpit

Die Prüfvariante ABAP\_CLOUD\_READINESS unterteilt sich in vier Unterbereiche (siehe Tabelle 1.1).

Bezeichnung	Name der Prüfung	Beschreibung
ABAP Language Version Check	SLIN_VERS	Prüfung, ob die Befehle in der Sprachversion ABAP Cloud genutzt werden dürfen
ABAP Language Version of Objects	SYCM_CHECK_ABAP_LANGU_VERSION	Prüfung, ob die Sprachversion des Objekts gleich der Softwarekomponente ist
Allowed Object Types in Cloud Development	SYCM_CLOUD_ALLOWED_OBJCT_TYPES	Prüfung, ob der Objekttyp für die ABAP Cloud unterstützt wird
Usage of Released APIs	SYCM_CLOUD_RELEASED_OBJECTS	Prüfung, ob die genutzten Objekte im Status RELEASED sind

**Tabelle 1.1** Prüfvariante ABAP\_CLOUD\_READINESS

Um die Qualität und Richtigkeit des Quellcodes zu prüfen, wird die Prüfvariante »ABAP\_CLOUD\_DEVELOPMENT\_DEFAULT« verwendet. Diese Prüfvariante führt die Unit-Tests für die Objekte aus und bestimmt die Qualität des Codings.

### SAP Cloud ALM

SAP Cloud ALM ist ein cloud-basierter Service, der von SAP auf der SAP BTP bereitgestellt wird. Es unterstützt den gesamten Application-Lifecycle-Management-Prozess für Cloud- und hybride SAP-Landschaften. SAP Cloud ALM wurde für folgende Funktionen entwickelt:

- Implementierung
- Betrieb
- Service Management
- Task- und Projektmanagement

Die Auslieferung der Clean Core Quality Gates erfolgt über SAP-Roadmaps. Clean Core Quality Gates sind ein Thema für das Gesamtprojekt, nicht nur für die Entwicklung. Quality Gates umfassen phasenspezifische Checklisten, die zu den jeweiligen Projektmeilensteinen bearbeitet werden müssen. Die Quality Gates sind über die Kachel **Task** und dort über die Auswahl des Task-Typs **Quality Gate** zugänglich (siehe Abbildung 1.10).

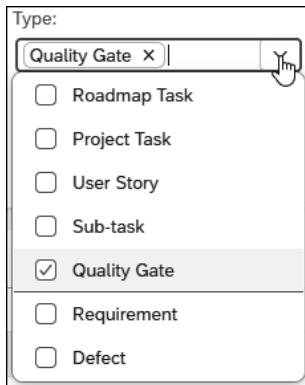


Abbildung 1.10 Task-Typen in SAP Cloud ALM

Die Bearbeitung der Clean Core Quality Gates erfolgt überwiegend durch die Beantwortung von Multiple-Choice-Fragen, die bei Bedarf durch ergänzende Kommentare präzisiert werden können. Jedes Clean Core Quality Gate besteht aus mehreren Elementen, die durch Setzen des Status **Accepted** abgeschlossen werden. Wenn alle einzelnen Elemente des Clean Core Quality Gates den Status **Accepted** oder **Conditionally Accepted** erreicht haben, kann das gesamte Quality Gate auf den Status **Accepted** gesetzt werden. Erst nachdem das gesamte Quality Gate auf **Accepted** gesetzt wurde, kann es über **Submit** an SAP gesendet werden.

## 1.2 Entwicklungsmöglichkeiten

Generell spricht man von drei Arten von Entwicklungen im SAP-Umfeld, die sich jeweils in ihrem Funktionsumfang unterscheiden:

- Key-User Extensions
- Cloud/On-Stack
- Side-by-Side

Während Key-User Extensions primär für kleinere Anpassungen gedacht sind, bieten Cloud/On-Stack- und Side-by-Side-Entwicklungen nahezu unbegrenzte Entwicklungsmöglichkeiten und sind in der SAP BTP umsetzbar. Tabelle 1.2 stellt diese Entwicklungsmöglichkeiten systematisch gegenüber und beleuchtet ihre spezifischen Charakteristika hinsichtlich Voraussetzungen, Zielgruppen, Anwendungsszenarien, Entwicklungsumgebungen und zeigt deren jeweilige Vor- und Nachteile.

	Key-User Extensions	Side-by-Side Extensions	Cloud/On-Stack
<b>Voraussetzung</b>	SAP-S/4HANA-System	SAP BTP und beliebige weitere Systeme (SAP ECC, SAP S/4HANA)	Drei-System-Landschaft von SAP S/4HANA Cloud oder ab Release 2022
<b>Zielgruppe</b>	Key-User und ABAP-Entwickler*innen	Entwickler*innen (Java-Script, Java etc.)	ABAP-Entwickler*innen
<b>Anwendbarkeit</b>	Erstellung einfacher Erweiterungen, etwa Hinzufügen von Feldern	Erstellung beliebig komplexer Anwendungen, die unabhängig von SAP S/4HANA skalieren und Personen ohne SAP-User bedienen können	Erstellung von Anwendungen/Erweiterungen, die stark mit SAP S/4HANA verbunden sind und das System direkt beeinflussen
<b>Entwicklungs-umgebung</b>	unterschiedliche Apps	SAP Build und weitere Entwicklungsservices	Eclipse
<b>Vorteile</b>	<ul style="list-style-type: none"> <li>■ Erweiterungen können schnell umgesetzt werden.</li> <li>■ einfach von Key-Usern durchführbar</li> <li>■ Erweiterungen ohne umfassende Entwicklungskenntnisse möglich</li> </ul>	<ul style="list-style-type: none"> <li>■ Anwendung, Datenbank skaliert (mehrere Instanzen können erzeugt werden, oder zusätzliche Ressourcen werden zugewiesen)</li> <li>■ Veröffentlichung als SaaS-Service möglich</li> <li>■ externe Nutzung außerhalb des Firmennetzes</li> <li>■ hohe Upgrade-Fähigkeit des Systems</li> </ul>	<ul style="list-style-type: none"> <li>■ Erweiterung ohne zusätzliche Systeme</li> <li>■ hohe Upgrade-Fähigkeit des Systems</li> </ul>
<b>Nachteile</b>	eingeschränkte Erweiterbarkeit	ggf. laufende Kosten durch die SAP BTP	Anwendung nicht skalierbar, nicht als SaaS möglich

Tabelle 1.2 Vergleich der Entwicklungsplattformen

### Richtige Entwicklungsplattform

Optimale Entwicklungsergebnisse werden erzielt, wenn man sich nicht auf eine einzelne Entwicklungsplattform festlegt. Entscheidend sind die individuelle Betrachtung jedes Use Cases und die darauf basierende Auswahl der geeigneten Entwicklungsplattform.



Nach der detaillierten Betrachtung der Entwicklungsplattform gilt es, für jeden spezifischen Use Case die passende Plattform zu identifizieren. Abbildung 1.11 visualisiert einen Entscheidungsbaum, der diesen Auswahlprozess systematisch unterstützt. Der Entscheidungsbaum untergliedert sich in fünf zentrale Kategorien:

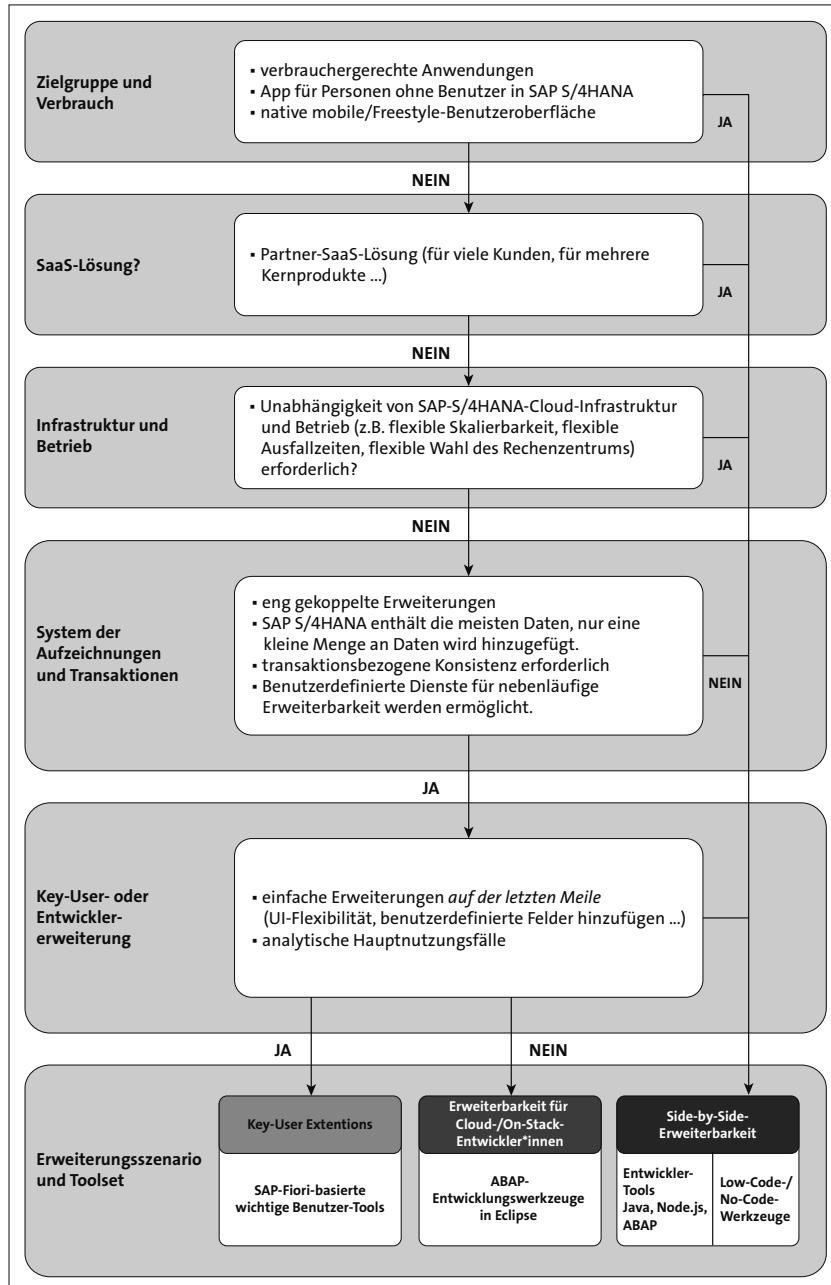


Abbildung 1.11 Entscheidungsbaum

1. Zielgruppe und Verbrauch
2. SaaS-Lösung
3. Infrastruktur und Betrieb
4. System der Aufzeichnung und Transaktionen
5. Key-User- oder Entwickler-Erweiterung

Jede Kategorie umfasst spezifische Entscheidungskriterien, die eine differenzierte Bewertung ermöglichen.

Folgendes Beispiel zeigt einen klassischen Use Case. Die Systemlandschaft umfasst mehrere SAP-S/4HANA-Systeme, da verschiedene Landesgesellschaften eigene Systeminstanzen benötigen. Es wird eine App benötigt, mit der SAP-Benutzer den Lagerbestand zu einem Artikel weltweit prüfen müssen.

Mit dem Use Case kann man nun den Entscheidungsbaum durchlaufen:

1. **Zielgruppe und Verbrauch:** Wird die App als mobile App benötigt? Sollen Nicht-SAP-Anwender\*innen Zugriff haben?

Nein, die Anwendung wird nicht als mobile App benötigt, und alle Anwender\*innen haben eine SAP-Lizenz.

2. **SaaS-Lösung?:** Soll die Anwendung als SaaS-Lösung entwickelt werden?

Nein, es soll keine Partner- oder SaaS-Lösung für andere Kunden entwickelt werden.

3. **Infrastruktur und Betrieb:** Ist Unabhängigkeit von SAP S/4HANA Cloud notwendig?

Ja, es wird die Unabhängigkeit von der SAP-S/4HANA-Cloud-Infrastruktur benötigt, da mehrere Systeme abgefragt werden müssen.

Für unseren Beispielfall lautet das Ergebnis demnach: Side-by-Side-Entwicklung.

## 1.3 Fünf Säulen der SAP BTP

Die SAP Business Technology Platform basiert auf fünf strategischen Säulen, die das Leistungsspektrum der Plattform strukturieren:

- Anwendungsentwicklung
- Automatisierung
- Integration
- Daten und Analytics
- KI

In manchen Darstellungen werden inzwischen Application Development und Automation zu einer gemeinsamen Säule zusammengefasst. Abbildung 1.12 gibt Ihnen einen Überblick über die SAP Business Technology Platform. Ergänzend zu den fünf Säulen umfasst die SAP BTP noch die übergreifenden Bereiche *Discovery Center* und *Ökosystem und Marketplace*.

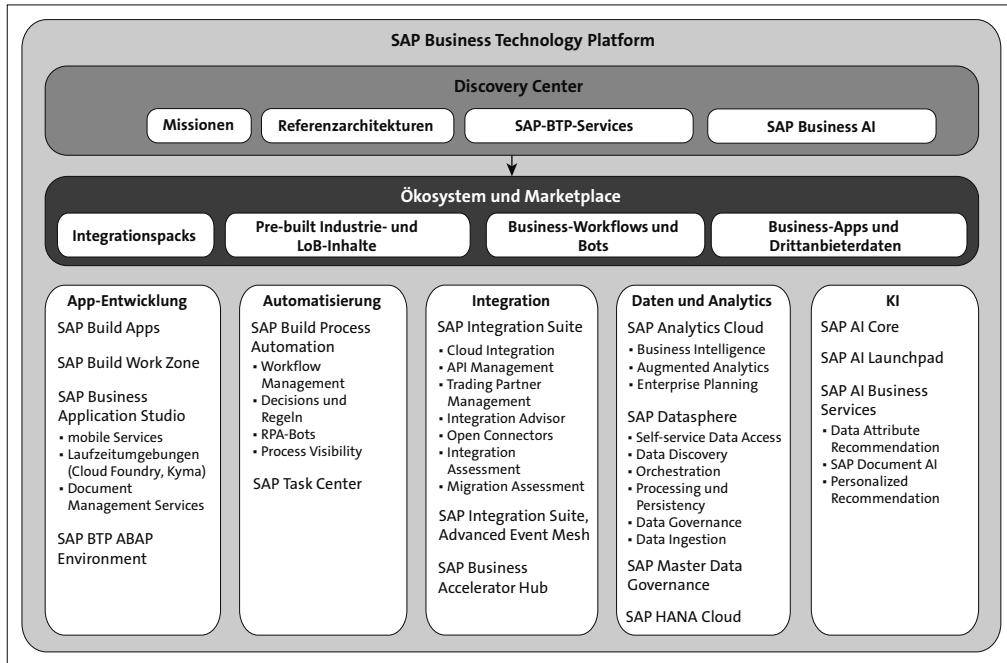


Abbildung 1.12 SAP Business Technology Platform

### 1.3.1 Integration

Die Integration moderner SAP-Systemlandschaften ist durch eine hohe Systemvielfalt gekennzeichnet: Cloud-Lösungen, On-Premise-Systeme, Drittanbieter und IoT-Geräte müssen nahtlos zusammenarbeiten. Hybride Architekturen erschweren dabei die Datenkonsistenz und Prozessharmonisierung. Unterschiedliche Schnittstellenstandards, Update-Zyklen und Protokolle erhöhen die Komplexität. Gleichzeitig steigen die Anforderungen an Sicherheit, Datenschutz und Compliance – insbesondere bei der Cloud-Anbindung. Fehlende Standardisierung führt häufig zu individuellem Customizing mit hohem Wartungsaufwand. Diese Herausforderungen erfordern einen strukturierten, sicheren und skalierbaren Integrationsansatz.

#### SAP Integration Suite

Die *SAP Integration Suite* ist eine cloud-basierte, KI-gestützte Integrationsplattform (Integration Platform as a Service, iPaaS), die Unternehmen dabei unterstützt, so-

wohl SAP- als auch Nicht-SAP-Systeme effizient zu vernetzen. Sie ermöglicht die nahtlose Integration von Cloud- und On-Premise-Lösungen sowie von Drittanbietersystemen innerhalb einer zentralen Plattform. Ziel ist es, Geschäftsprozesse durchgängig zu gestalten und manuelle Zwischenschritte und inkonsistente Datenformate zwischen verschiedenen Systemen zu vermeiden (sogenannte *Medienbrüche*), so dass der Datentransfer automatisiert, durchgängig und fehlerfrei abläuft. Dabei bietet die SAP Integration Suite mehr als 10.500 vorkonfigurierte Integrationsinhalte – darunter Konnektoren, Mapping-Vorlagen und branchenspezifische Best Practices – die eine schnelle und standardisierte Umsetzung von Integrationsszenarien ermöglichen.

### Cloud Integration

Cloud Integration, früher bekannt als SAP Cloud Platform Integration (CPI), ist eine cloud-basierte Integrationsplattform, die es Unternehmen ermöglicht, verschiedene Anwendungen und Systeme zu verbinden und zu integrieren, sowohl als On-Premise-Lösung als auch in der Cloud. Sie erleichtert den nahtlosen Datenfluss und die Prozesskonnektivität zwischen verschiedenen Systemen. Sie ist eine Kernfunktion innerhalb der SAP Integration Suite, die wiederum Teil der SAP BTP ist. Cloud Integration ist eine Platform-as-a-Service-Lösung, die in der Cloud gehostet wird und Skalierbarkeit und Zugänglichkeit bietet. Sie unterstützt eine breite Palette von Integrationsszenarien, darunter:

- Verbindung von On-Premise- und Cloud-Anwendungen
- Integration von hybriden und Cloud-System-Landschaften
- intuitive Weboberfläche mit integrierten Vorlagen
- Unterstützung verschiedener Szenarien, wie z. B. Application to Application (A2A), Business to Business (B2B) und Business to Government (B2G) mit Schwerpunkt auf Nicht-SAP-Anwendungen
- Cloud-Technologie
- starker Sicherheitsfokus und hohe Verfügbarkeit der Infrastruktur
- Erweiterbarkeit der vorgefertigten Integrationsinhalte und deren Nutzung

### API Management

Ein effektives API Management ist essenziell, um moderne Anwendungen und Integrationen erfolgreich umzusetzen. Dabei geht es nicht nur darum, APIs sicher bereitzustellen, sondern auch darum, sie effizient zu verwalten und bei Bedarf zu skalieren. Unternehmen können so Daten und Funktionen aus SAP- sowie Drittanwendungen strukturiert und kontrolliert nach außen freigeben. Die Nutzung dieser APIs lässt sich über Dashboards und Analysen transparent überwachen, was eine kontinuierliche Optimierung ermöglicht. Der Zugriff erfolgt dabei präzise gesteuert über Rollen,

Richtlinien und moderne Sicherheitsstandards wie OAuth2. APIs fungieren somit als digitale Bausteine für innovative Applikationen und offene Integrationsszenarien. Ein durchdachtes API Management unterstützt den gesamten Lebenszyklus dieser Schnittstellen – von der Entwicklung über die Veröffentlichung bis hin zur Überwachung und Weiterentwicklung. Die folgende Liste spiegelt die Kernfunktionen des API Managements wider.

1. API-Design und -Veröffentlichung
  - Erstellung und Pflege von APIs über ein zentrales Cockpit
  - Versionierung und Dokumentation
2. Zugriffskontrolle und Sicherheit
  - Policies, Token, Quotas und Zugriffsbeschränkungen
  - Unterstützung für OAuth, SAML, HTTPS etc.
3. Überwachung und Analyse
  - Echtzeit-Dashboards zur API-Nutzung
  - Fehlertracking, Aufrufzahlen, Antwortzeiten
4. Entwicklerportal
  - eigene APIs für interne oder externe Entwickler\*innen zugänglich machen
  - Dokumentation, Testfunktion, Self-Service-Zugriff

### **Trading Partner Management**

Trading Partner Management ist eine leistungsstarke Funktion innerhalb der SAP Integration Suite, die die Verwaltung und Steuerung von B2B-Beziehungen, Vereinbarungen und Interaktionen mit externen Handelspartnern vereinfacht. Mit Trading Partner Management können Unternehmen ihren B2B-Datenaustausch auf ihre spezifischen Anforderungen zuschneiden, indem sie für jeden Handelspartner Protokolle, Standards und APIs für den elektronischen Datenaustausch definieren und konfigurieren.

### **Integration Advisor**

Integration Advisor ist eine Cloud-Anwendung, die Ihnen hilft, den Implementierungsablauf Ihres B2B/A2A- und B2G-Integrationsprozesses zu vereinfachen und zu optimieren. Sie nutzt einen crowd-basierten Ansatz für maschinelles Lernen, um Sie bei der einfachen Erstellung von Integrationsinhalten zu unterstützen. Der Integration Advisor unterstützt die KI-gestützte Erstellung von Nachrichtenformaten, Mappings und zugehörigen Laufzeitartefakten inklusive Dokumentation. Dabei greift er auf eine bereitgestellte Bibliothek mit gängigen EDI-Standardformaten zurück.

## Open Connectors

Open Connectors ist ein zentraler Bestandteil der SAP Integration Suite und bietet standardisierte, sofort nutzbare Konnektoren für über 160 Anwendungen von Drittanbietern. Ziel ist es, die Integration externer Cloud-Services – wie Salesforce, Slack, Stripe oder Google Drive – mit SAP-Systemen deutlich zu vereinfachen. Dafür stellt Open Connectors ein einheitliches REST-API-Format bereit, das Unterschiede in Authentifizierung, Datenmodell und Schnittstellentechnologie abstrahiert. Unternehmen können so externe Services schneller und sicherer in ihre Geschäftsprozesse einbinden, ohne die spezifische Logik oder Komplexität jeder einzelnen Drittanbieter-API selbst implementieren zu müssen.

## Integration Assessment

Integration Assessment implementiert die Integration Solution Advisory Methodology und unterstützt damit die Definition der Integrationslandschaft. Es hilft Ihnen bei der Definition von Integrationsmustern und bei der Sicherstellung einer konsistenten Nutzung der Integrationstechnologie. Integration Assessment bietet Ihnen einen geführten Ansatz zur Bewertung Ihrer Integrationsstrategie. Es hilft Ihnen, Empfehlungen zu den Integrationstechnologien zu erhalten, die für die Anforderungen Ihres Unternehmens geeignet sind.

## SAP Integration Suite, Advanced Event Mesh

Advanced Event Mesh ist ein vollständig verwalteter Cloud-Service innerhalb der SAP Integration Suite, der es Unternehmen ermöglicht, eine skalierbare und ereignisgesteuerte Architektur (engl. Event-Driven Architecture, EDA) aufzubauen und zu betreiben. Der Service stellt eine zuverlässige, asynchrone Kommunikation zwischen Anwendungen, Diensten und Systemen sicher – sowohl in hybriden als auch in Multi-Cloud- und Edge-Umgebungen.

Durch die Verwendung standardisierter Messaging-Protokolle (z. B. AMQP, MQTT, REST, WebSocket) können sowohl SAP- als auch Drittanbietersysteme nahtlos integriert werden. Geschäftliche Ereignisse lassen sich in Echtzeit veröffentlichen, abonnieren, filtern und gezielt weiterleiten. Unternehmen profitieren dadurch von einer entkoppelten Systemarchitektur, die eine flexible Skalierung, reduzierte Abhängigkeiten und eine höhere Ausfallsicherheit ermöglicht.

Zusätzlich bietet Advanced Event Mesh umfassende Funktionen für die zentrale Überwachung, Steuerung und Analyse von Event-Flüssen, was eine transparente und nachvollziehbare Orchestrierung unternehmensweiter Prozesse erlaubt.

## SAP Business Accelerator Hub

Der SAP Business Accelerator Hub (<https://api.sap.com>) ist die zentrale Plattform für Entwickler\*innen, um auf standardisierte SAP APIs, Integrationspakete und Beispielanwendungen zuzugreifen. Er bietet eine umfassende Sammlung von REST-, OData- und SOAP-Schnittstellen für SAP S/4HANA, SAP SuccessFactors, SAP Ariba, SAP BTP sowie Drittanbietersysteme. Nutzer\*innen können APIs direkt im Browser durchsuchen, dokumentierte Endpunkte testen und mit Sandbox-Daten simulieren – inklusive Codebeispielen und Authentifizierungsoptionen. Die Plattform unterstützt eine schnelle und effiziente Anbindung von SAP- und Nicht-SAP-Systemen und erleichtert so die Umsetzung moderner, API-basierter Integrationsszenarien.

Abbildung 1.13 veranschaulicht den Aufbau einer Systemlandschaft und zeigt, welche Integrationsszenarien auftreten und mit der SAP Integration Suite abgewickelt werden können.

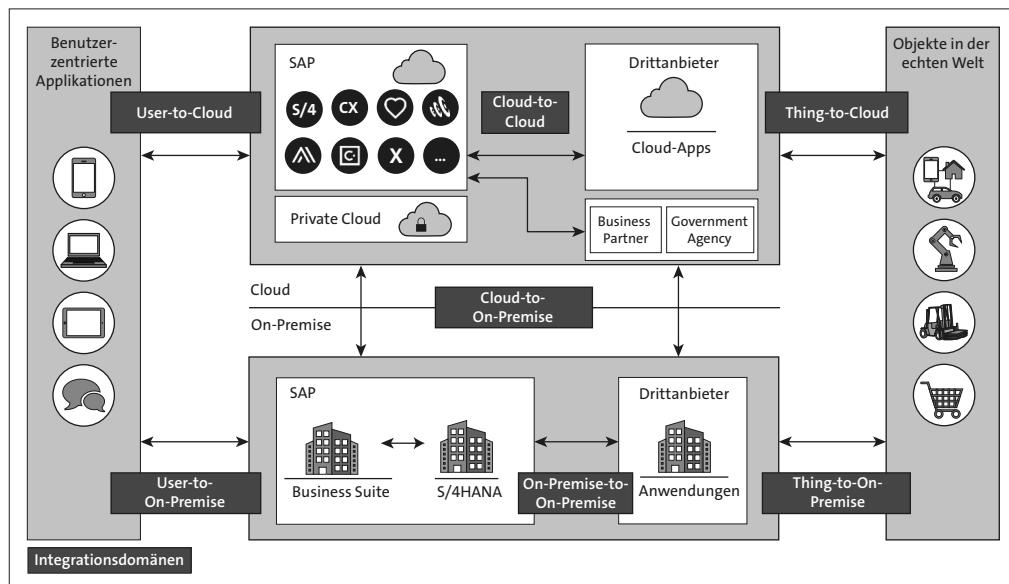


Abbildung 1.13 SAP Integration Suite

### 1.3.2 Daten und Analytics

Im Bereich Daten und Analytics umfasst die SAP BTP mehrere Kernkomponenten, die gemeinsam eine skalierbare und KI-fähige Datenarchitektur ermöglichen:

- SAP Analytics Cloud
- SAP Datasphere
- SAP Master Data Governance
- SAP HANA Cloud
- SAP Business Data Cloud

### SAP Analytics Cloud

Die SAP Analytics Cloud ist die zentrale Lösung für Business Intelligence, Planung und vorausschauende Analysen. Sie basiert auf einer In-Memory-Architektur und bietet eine vollständig integrierte Umgebung für:

- **Self-Service BI:** Benutzer können interaktive Dashboards und Visualisierungen erstellen.
- **Enterprise Planning:** Integrierte Planungsfunktionen ermöglichen Szenarioanalysen, Forecasting und Budgetierung.
- **Augmented Analytics:** KI-gestützte Funktionen wie Smart Insights, Smart Discovery und Natural Language Querying (NLQ) beschleunigen die Erkenntnisgewinnung.

SAP Analytics Cloud ist vollständig in SAP Datasphere und SAP Business Data Cloud integriert und unterstützt sowohl SAP- als auch Nicht-SAP-Datenquellen.

### SAP Datasphere

SAP Datasphere ist die strategische Weiterentwicklung von SAP Data Warehouse Cloud. Sie ermöglicht eine Datenvirtualisierung und Integration über verschiedene Quellsysteme hinweg ohne physische Datenreplikation. Zu den zentralen Funktionen gehören:

- **Semantische Modellierung:** Daten werden, basierend auf dem SAP One Domain Model, mit Geschäftskontext angereichert.
- **Datenföderation:** Zugriff auf verteilte Datenquellen in Echtzeit ohne Duplizierung
- **Data Catalog und Lineage:** Metadatenmanagement, Datenherkunft und Impact-Analysen
- **Spaces und Domains:** logische Trennung von Datenbereichen zur Umsetzung von Data-Mesh-Architekturen
- **Integration mit SAP HANA Cloud:** für performante In-Memory-Verarbeitung und persistente Speicherung

SAP Datasphere ist die technische Grundlage für die Umsetzung von Data-Mesh-Prinzipien in SAP-Umgebungen.

### SAP Master Data Governance (SAP MDG)

SAP Master Data Governance (SAP MDG) stellt sicher, dass Stammdaten unternehmensweit konsistent, qualitätsgesichert und regelkonform verwaltet werden. Die Lösung unterstützt:

- zentrale Datenpflegeprozesse mit Workflows, Validierungen und Dublettenprüfung
- domänenübergreifende Stammdatenmodelle (z. B. Business Partner, Material, Finanzobjekte)
- Integration mit SAP S/4HANA und SAP Datasphere zur Sicherstellung konsistenter Analysen und operativer Prozesse

### SAP HANA Cloud

SAP HANA Cloud ist eine In-Memory-Datenbankplattform. Sie bietet:

- hybridfähige Datenverarbeitung (transaktional und analytisch, OLTP – Online Transactional Processing und OLAP – Online Analytical Processing)
- native Unterstützung für SQL, Graph, Spatial, JSON und Zeitreihen
- Data-Lake-Integration für die kosteneffiziente Speicherung großer Datenmengen
- Machine-Learning-Integration über SAP HANA Machine Learning und Python-/R-Schnittstellen

SAP HANA Cloud dient als performante Persistenzschicht für SAP Datasphere und ist vollständig in die SAP BTP eingebettet.

### SAP Business Data Cloud

SAP Business Data Cloud ist die neueste Innovation im SAP-Portfolio und vereint die Funktionen von SAP Datasphere, SAP Analytics Cloud, SAP Business Warehouse und SAP Databricks in einer einheitlichen SaaS-Plattform. Ihre Schlüsselmerkmale sind:

- **Kurierte Datenprodukte:** Vorgefertigte, semantisch angereicherte Datensätze auf Basis des SAP One Domain Models. Diese Produkte sind versioniert, dokumentiert und über den Data Catalog auffindbar.
- **Intelligente Anwendungen:** KI-gestützte, domänenspezifische Analyse-Apps, die direkt auf die Datenprodukte zugreifen und in Echtzeit Insights liefern
- **Zero Copy Sharing:** Daten werden nicht repliziert, sondern über föderierte Zugriffe bereitgestellt – mit vollständiger Datenherkunft und Auditierbarkeit.
- **Wissensgraphen und semantische Modellierung:** Verknüpfung von Daten, Metadaten und Prozessen zur Unterstützung von KI und automatisierten Entscheidungen
- **Joule:** ein generativer KI-Assistent, der Aufgaben automatisiert, Insights generiert und die Interaktion mit Daten vereinfacht

### 1.3.3 Künstliche Intelligenz

Die Entwicklung begann mit klassischen Machine-Learning-Ansätzen zur Automatisierung von Rechnungsprozessen, bei denen das System lernte, Zahlungen automa-

tisch den richtigen Rechnungen zuzuordnen. Diese frühen Anwendungen legten den Grundstein für datengetriebene Entscheidungen und kontinuierliches Lernen aus Benutzerinteraktionen. Mit dem technologischen Fortschritt verlagerte sich der Fokus zunehmend auf generative KI. Künstliche Intelligenz (KI) ist längst mehr als nur ein vorübergehender Trend, sie entwickelt sich zunehmend zu einem wichtigen Bestandteil im Alltag. Unternehmen setzen verstärkt auf KI-Technologien zur Prozessoptimierung und Effizienzsteigerung. Die SAP BTP stellt dafür leistungsstarke Werkzeuge zur Verfügung:

- SAP AI Core: Bildet das Fundament für die Entwicklung und den Betrieb von KI-Modellen.
- SAP AI Launchpad: Ermöglicht die zentrale Verwaltung eigenentwickelter und von Drittanbietern entwickelter Modelle.

Diese erlauben es Entwickler\*innen, eigene KI-Modelle zu trainieren, zu verwalten und produktiv bereitzustellen. Ein weiteres Feld dieser Entwicklung ist SAP Business AI, eine neue Generation von KI-Diensten, die tief in SAP-Produkte wie SAP S/4HANA oder SAP SuccessFactors eingebettet ist. Im Mittelpunkt dieser Strategie steht Joule, der generative KI-Assistent von SAP. Joule basiert auf Large Language Models (LLMs), ist jedoch nicht nur ein Sprachmodell, sondern tief mit Geschäftslogik und Stammdaten verknüpft. Dadurch kann Joule nicht nur generische Antworten liefern, sondern kontextbezogene Einblicke und Empfehlungen geben. Entwickler nutzen Tools wie SAP Build Code oder das SAP Business Application Studio, um Anwendungen zu entwickeln und KI-Funktionen zu integrieren. Neben SAP-eigenen Modellen können auch externe KI-Dienste über APIs eingebunden werden. Joule selbst ist in drei Varianten verfügbar:

- Joule Business AI
- SAP Joule for Consultants
- SAP Joule for Developers

Darüber hinaus bietet SAP eine Reihe von vorkonfigurierten AI Services, die *SAP AI Services*, die sich in bestehende Geschäftsprozesse und Entwicklungen integrieren lassen, hier eine Auswahl der Services:

- SAP Document AI: automatisierte Informationen aus strukturierten und unstrukturierten Dokumenten extrahieren
- Data Attribute Recommendation: liefert intelligente Vorschläge für fehlende Attribute, z. B. bei der Stammdatenpflege
- Personalized Recommendation – erstellt Produktempfehlungen und verbessert die Benutzererfahrung der Kunden

Diese Services können von Unternehmen auch ohne umfassende Data-Science-Kenntnisse für Automatisierungen von Routineaufgaben oder zur Umsetzung eigener Lösungen genutzt werden. Mit *Joule* wird diese Integration noch einfacher. Joule versteht natürliche Sprache, liefert kontextbezogene Antworten und unterstützt Mitarbeitende direkt in ihren täglichen Anwendungen.

Abbildung 1.14 zeigt einmal den Überblick der SAP AI Services auf der SAP BTP.

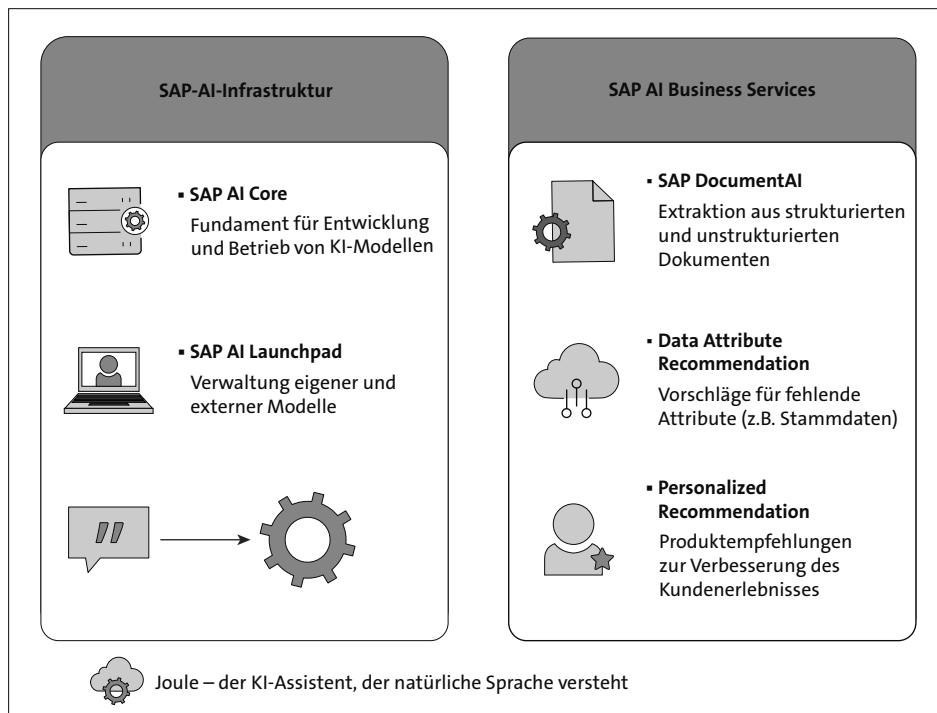


Abbildung 1.14 Künstliche Intelligenz auf der SAP BTP

Die Integration dieser KI-Services in die Entwicklung erfolgt über verschiedene Wege. Entwickler\*innen können KI-Modelle direkt über das Launchpad in ihre Anwendungen einbinden, etwa über REST APIs. Das SAP Cloud Application Programming Model bietet eine moderne Architektur für die Entwicklung cloud-nativer Anwendungen mit KI. Auch ABAP in der Cloud kann KI-Funktionen nutzen. SAP Build ermöglicht es, KI-gestützte Anwendungen zu erstellen. Ein weiterer technologischer Meilenstein ist die Integration der Vector Engine in die SAP HANA Cloud. Diese ermöglicht semantische Suchen und Ähnlichkeitsanalysen, was essenziell für moderne LLM-Anwendungen ist, bei denen Informationen effizient verarbeitet werden müssen.

Ausführliche Informationen zum Thema KI finden Sie in Kapitel 6, »Künstliche Intelligenz auf der SAP BTP«.

### 1.3.4 Automatisierung

Die Kombination aus SAP Build Process Automation und SAP Signavio bietet Unternehmen eine integrierte Plattform, mit der sie ihre End-to-End-Geschäftsprozesse modellieren, automatisieren und optimieren können. Durch die enge Zusammenarbeit von Prozessdesign, -ausführung und -analyse entsteht ein durchgängiger Regelkreis. Dadurch werden die Prozessflüsse transparenter, und es werden schneller Automatisierungspotenziale erkannt. Der Fokus liegt auf der Implementierung, Fehlervermeidung und Qualitätssicherung durch den Einsatz von Low-Code-/No-Code-Technologien. So können Workflows, Logik und Automatisierungen mit direkter Anbindung an bestehende SAP- und Nicht-SAP-Systeme über Schnittstellen und Events umgesetzt werden.

SAP Build Process Automation bildet dabei die zentrale Ausführungs- und Orchestrierungsschicht. Dies umfasst die folgenden Funktionen:

- Workflow Management
- Anlegen von Entscheidungen (engl. Decisions) und Regeln (engl. Rules)
- Entwicklung von Robotic Process Automation Bots
- Process Visibility

Ergänzt wird die Automatisierung durch das SAP Task Center, das als zentrale Inbox für Interaktionen dient. Es leitet Aufgaben aus verschiedenen Backend-Systemen zusammen und stellt sie den Benutzer\*innen gesammelt zur Verfügung.

Ausführliche Informationen zum Thema Automatisierung finden Sie in Kapitel 4, »SAP Build«.

### 1.3.5 Anwendungsentwicklung

Die SAP BTP bildet die Grundlage für die Entwicklung von Geschäftsanwendungen. Sie bietet eine große Auswahl an Technologien, Frameworks und Werkzeugen, mit denen sich Anwendungen entwickeln lassen. Folgende Entwicklungsmöglichkeiten werden bereitgestellt:

- SAP Build Apps: ein benutzerfreundliches Tool für die No-Code- und Low-Code-Entwicklung
- SAP Build Work Zone: Die Lösung ermöglicht die Gestaltung benutzerfreundlicher digitaler Arbeitsbereiche, in denen Anwendungen, Inhalte und Prozesse zentral zusammengeführt werden.
- SAP Business Application Studio: eine cloud-basierte Entwicklungsumgebung für professionelle Entwickler\*innen
- SAP BTP ABAP Environment: eine cloud-native Umgebung für die Entwicklung mit ABAP

Zusätzlich werden noch weitere Services für den Betrieb und verschiedene Anwendungsfälle bereitgestellt:

- SAP Mobile Services: unterstützen die Entwicklung mobiler Anwendungen mit Funktionen wie Push-Benachrichtigungen, Offline-Fähigkeit und Geräteintegration
- *Cloud-Laufzeitumgebungen* wie Cloud Foundry und Kyma: Sie ermöglichen es, Applikationen und Microservices flexibel laufen zu lassen.
- SAP Document Management Service: ermöglicht die strukturierte Verwaltung, Speicherung und Integration von Dokumenten in der SAP BTP

Die SAP BTP unterstützt außerdem das SAP Cloud Application Programming Model (CAP), ABAP RESTful Application Programming Model und SAP Build in der Entwicklung.

Damit bietet die SAP BTP ein leistungsstarkes Fundament für die Entwicklung zukunftsfähiger Cloud-Anwendungen. Mit CAP können Anwendungen schnell modelliert werden – von der Datenbank bis zur Benutzeroberfläche. Das Framework ermöglicht die automatische API-Generierung und folgt dem MVC Pattern (Model View Controller) mit klarer Trennung von Daten, Services und Logik.

ABAP RESTful Application Programming Model bringt ABAP in die Cloud und ermöglicht die Entwicklung von Business-Logik auf Basis von Core Data Services und OData-Services.

SAP Build erweitert die Entwicklungsmöglichkeiten um ein No-Code-, Low-Code- und Pro-Code-Portfolio, um Prozesse zu automatisieren, Benutzeroberflächen zu gestalten und Anwendungen schnell bereitzustellen.

Ob Erweiterung von SAP S/4HANA, Side-by-Side-Apps oder Microservices, die SAP BTP bietet mit diesen Technologien die Plattform für Ihre SAP-spezifischen Entwicklungen. Zusätzlich werden Entwickler\*innen mit KI-Technologie wie Joule bei ihrer Arbeit unterstützt.

### SAP Cloud Application Programming Model

Das SAP Cloud Application Programming Model (CAP) ist ein modernes Framework zur Entwicklung von Applikationen und Services in der Cloud. Es wurde speziell für die SAP BTP entwickelt und unterstützt bei der Entwicklung von cloud-basierten Applikationen, die als Standalone-Applikationen oder als Side-by-Side Extensions genutzt werden können. Im Kern von CAP steht ein serviceorientierter Ansatz, bei dem definierte Schnittstellen den Zugriff auf Daten und Funktionen bereitstellen.

Das Ziel von CAP besteht darin, die Entwicklung effizienter und konsistenter zu gestalten. Dafür nutzt es Best Practices, vorkonfigurierte Werkzeuge und eine Architektur, die alle Schichten von der Datenbank bis zur Benutzeroberfläche abdeckt. Die Entwicklung ist sowohl lokal als auch in der Cloud möglich. Im Mittelpunkt steht dabei die deklarative Modellierung mit Core Data Services (CDS). Mit dieser gut lesbaren, domänen spezifischen Sprache lassen sich Datenmodelle und Services so beschreiben, wie sie in der jeweiligen Geschäftswelt wirklich verstanden werden. Der große Vorteil ist, dass aus diesen Modellen automatisch APIs, Benutzeroberflächen und Datenbankstrukturen generiert werden können. Das spart nicht nur Zeit, sondern reduziert auch typische Fehlerquellen. Die sogenannte *Serviceschicht* stellt diese Modelle dann über bewährte Protokolle wie OData oder REST zur Verfügung. Mit diesen Schnittstellen können gezielte Entwicklungen mit zusätzlicher Geschäftslogik versehen werden, wie z. B.:

- Validierungen
- Datenaufbereitung
- Berechtigungsprüfungen

Durch die saubere Trennung von Datenmodell, Servicebeschreibung und Logik bleibt der Code übersichtlich, leichter wartbar und gut wiederverwendbar. CAP lässt bei der Wahl der Entwicklungsumgebung viel Freiheit. Neben klassischen Tools wie Visual Studio Code ist vor allem das SAP Business Application Studio zu nennen. Diese cloud-basierte Entwicklungsumgebung ist speziell für die SAP BTP optimiert und unterstützt mit vielen Tools, wie z. B.:

- CDS Graphical Modeler zur Visualisierung von Datenmodellen
- Page Map zur Konfiguration der Benutzeroberfläche
- integriertes Debugging

Ein echter Pluspunkt von CAP ist zudem seine Offenheit. Es lässt sich nahtlos mit anderen SAP-Services und auch mit externen APIs kombinieren. Unterstützt werden verschiedene Datenbanksysteme wie SAP HANA, SQLite oder PostgreSQL. Ob synchrone oder asynchrone Services, beides ist möglich. Dadurch eignet sich CAP für eine breite Palette von Anwendungsszenarien: von Erweiterungen für SAP S/4HANA über Side-by-Side-Apps bis hin zu Microservices-Architekturen. CAP bringt Struktur in komplexe Anwendungen, unterstützt moderne Entwicklungsstandards und schafft damit eine zukunfts-fähige Basis für die digitale Transformation im SAP-Kosmos. Abbildung 1.15 zeigt einen Überblick des SAP Cloud Application Programming Models, das in Kapitel 2 genauer betrachtet wird.

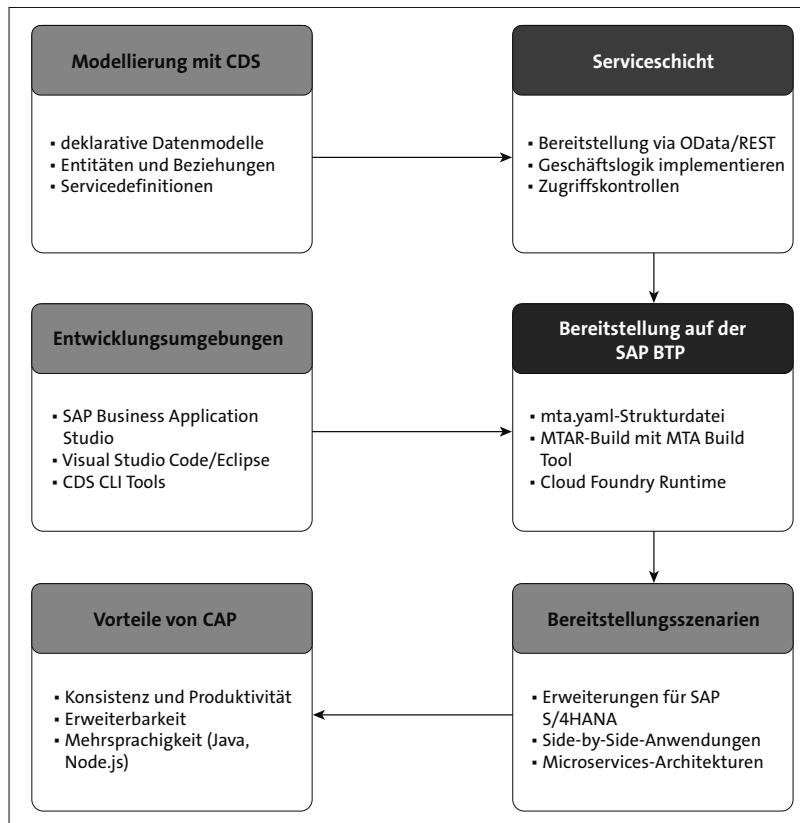


Abbildung 1.15 Überblick über das SAP Cloud Application Programming Model

### ABAP in der Cloud

ABAP in der Cloud steht für die nächste Generation in der ABAP-Entwicklung. Die ursprünglich strukturierte und dann später objektorientierte Sprache ABAP, die traditionell eng mit On-Premise-Systemen und direkten Datenbankzugriffen verknüpft war, bringt in ihrer Cloud-Variante neue Anforderungen, Konzepte und Werkzeuge mit sich. Diese Sprachvariante wurde initial für das ABAP Environment auf der SAP Business Technology Platform (auch als *Steampunk* bezeichnet) konzipiert und erlaubt neue Möglichkeiten der Entwicklung in SAP S/4HANA. Der Zugriff auf Daten erfolgt ausschließlich über freigegebene APIs, was die Sicherheit und Wartbarkeit erhöht. Ein wesentliches Merkmal der Cloud-Entwicklung ist die Nutzung der Sprachvariante *ABAP for Cloud Development*. Diese Version ist auf Stabilität und Update-Fähigkeit ausgelegt. Viele klassische Sprachelemente wie MOVE oder OPEN DATASET sind nicht mehr erlaubt, da sie entweder veraltet oder sicherheitskritisch sind. Auch SQL muss in moderner Syntax geschrieben werden. Die Nutzung von Systemobjekten ist nur über freigegebene APIs möglich, deren Stabilität durch SAP garantiert wird. Die Freigabestatus (CO bis C4) regeln, wie ein Objekt verwendet werden darf, als

API, für Konfigurationen oder in ABAP-Managed Procedures. Ein zentrales Entwicklungsmodell in der Cloud ist das ABAP RESTful Application Programming Model. Es basiert auf Core Data Services und dem Virtual Data Model (VDM), das in drei Schichten unterteilt ist.

- Basic Views (direkter Zugriff auf Tabellen)
- Composite Views (Kombination mehrerer Views mit Logik)
- Consumption Views (für konkrete Anwendungsfälle wie SAP-Fiori-Apps oder APIs)

Beziehungen zwischen Entitäten werden über Assoziationen und Kompositionen modelliert. Das ABAP RESTful Application Programming Model definiert nicht nur die Datenstruktur, sondern auch das Verhalten von Business Objects. Über sogenannte *Behavior Definitions* wird festgelegt, welche Operationen (Create, Update, Delete), Validierungen und Sperrmechanismen ein Objekt unterstützt. Die Implementierung erfolgt in ABAP-Klassen, die von speziellen Basisklassen erben. Die Runtime unterscheidet zwischen einer Interaktionsphase (z. B. Änderungen im Buffer) und einer Speicherphase (Persistierung auf der Datenbank). Ein weiteres zentrales Element ist die Entity Manipulation Language (EML). Sie ermöglicht den Zugriff auf Business Objects in ABAP, sowohl lesend als auch schreibend. Über die EML können mehrere Datensätze gleichzeitig verarbeitet, Fehler strukturiert behandelt und Aktionen wie COMMIT ENTITIES und ROLLBACK ENTITIES ausgeführt werden. Das ABAP RESTful Application Programming Model bietet darüber hinaus erweiterte Funktionen:

- Determinations (automatische Feldberechnungen)
- Validations (Konsistenzprüfungen)
- Actions (schreibende Sonderfunktionen)
- Functions (lesende Sonderfunktionen)

Diese werden in der Behavior Definition definiert und über EML aufgerufen. Ein weiteres wichtiges Konzept ist die *automatische Nummernvergabe* (Numbering). Diese kann entweder in der Interaktionsphase (early numbering) oder in der Speicherphase (late numbering) erfolgen, je nachdem, ob z. B. ein Nummernkreis verwendet wird. Ergänzt wird dies durch das *Draft-Konzept*, das es erlaubt, Entwurfsdaten temporär zu speichern. Dies ist besonders für SAP-Fiori-Apps relevant, da Anwender\*innen ihre Eingaben zwischenspeichern können, bevor sie diese final speichern. Zur Wiederverwendung von Business Objects dienen Projektionen, die bestimmte Felder und Operationen gezielt freigeben. Über Service Definitions und Service Bindings werden diese Objekte dann als OData-Services veröffentlicht und entweder als UI Service (z. B. für SAP Fiori) oder als Web API verwendet. Die Entwicklung erfolgt typischerweise in Eclipse mit den ABAP Development Tools. Diese Umgebung ersetzt die klassische SE80 und bietet moderne Werkzeuge für CDS, ABAP RESTful Application Programming Model und Debugging. Für den schnellen Einstieg steht der *Business Object Generator* zur Verfügung, der auf Basis einer Datenbanktabelle automatisch

ein vollständiges ABAP-RESTful-Modell samt Service erzeugt. Kapitel 3, »ABAP in der Cloud und ABAP RESTful Application Programming Model«, zeigt die Entwicklung einer Kontaktverwaltung mit zugehörigen Adressen. Dabei werden Datenbanktabellen modelliert, CDS Views erstellt, Behavior Definitions implementiert, Validierungen und Determinations ergänzt sowie eine SAP-Fiori-Oberfläche mit UI-Annotierungen gestaltet. Die Anwendung wird über einen OData-Service veröffentlicht und ist sofort einsatzbereit. ABAP in der Cloud ist nicht nur eine technologische Weiterentwicklung, sondern auch die Zukunft von ABAP. Es bringt mehr Struktur, Sicherheit und Wartbarkeit und eröffnet neue Möglichkeiten für moderne, cloud-basierte Geschäftsanwendungen. Abbildung 1.16 gibt einen guten Überblick über das ABAP RESTful Application Programming Model, das in Kapitel 3 genauer betrachtet wird.

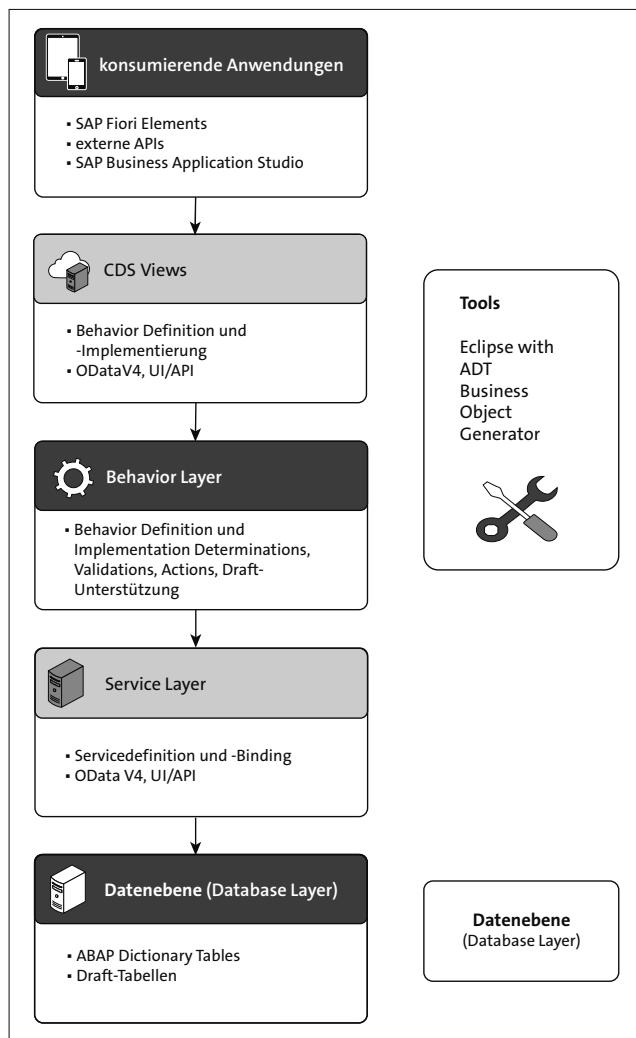


Abbildung 1.16 Überblick über das ABAP RESTful Application Programming Model

## SAP Build

Die SAP Build Suite ist eine Sammlung von Services innerhalb der SAP BTP, die Unternehmen bei der Entwicklung, Automatisierung und Verwaltung von Anwendungen unterstützt. Sie kombiniert folgende Konzepte:

- No-Code
- Low-Code
- Pro-Code

SAP Build richtet sich sowohl an Citizen Developer als auch an professionelle Entwickler\*innen. Die Suite besteht aus vier Komponenten:

- SAP Build Apps
- SAP Build Process Automation
- SAP Build Work Zone
- SAP Build Code

Diese Services ermöglichen es, Anwendungen schnell zu erstellen, Geschäftsprozesse zu automatisieren und eine zentrale Benutzeroberfläche für SAP- und Drittanwendungen bereitzustellen.

Die SAP-Build-Plattform bietet eine einfache Entwicklungsumgebung zur Unterstützung für Datenbindung, Logikflüsse und visuelle Komponenten sowie die Integration mit SAP- und Drittanbietersystemen. Es stehen verschiedene Variablenarten zur Verfügung. Der Theme Editor erlaubt die Anpassung des Designs, und eigene Komponenten können über den Marketplace hinzugefügt werden.

*SAP Build Process Automation* vereint Funktionen aus SAP Workflow Management und SAP Intelligent Robotic Process Automation. Sie bietet eine einfache Modellierung von Geschäftsprozessen über den Process Builder, die Erstellung von Formularen, Entscheidungen und Automatisierungen sowie die Integration von Bots und Business Rules. Bestehende Workflows können wiederverwendet und externe Systeme über Aktionen eingebunden werden. Die zentrale Verwaltung erfolgt über den Control Tower, während My Inbox und Monitoring für Transparenz im laufenden Betrieb sorgen.

*SAP Build Work Zone* ist als zentraler Einstiegspunkt in die SAP-Welt zu sehen. Sie ist in zwei Editionen verfügbar:

- Standard Edition (entspricht dem klassischen SAP Fiori Launchpad)
- Advanced Edition mit erweiterten Funktionen für:
  - soziale Netzwerke
  - Wissensdatenbanken
  - kollaborative Arbeitsbereiche

SAP Build Work Zone ermöglicht die Integration von SAP- und Drittanbieteranwendungen, bietet persönliche Arbeitsbereiche (My Workspace) und thematische Workspaces mit Widgets. Die zentrale Inbox erlaubt alle Workflows in einer App, und zusätzliche Tools wie Kalender, Empfehlungen und Lesezeichen stehen zur Verfügung. Die Integration mit Microsoft Teams erweitert SAP Build Work Zone zu einem modernen digitalen Arbeitsplatz.

*SAP Build Code* richtet sich an professionelle Entwickler\*innen und bietet eine moderne Entwicklungsumgebung mit Unterstützung für Node.js und Java. Die Integration von Joule, einem KI-gestützten Entwicklungsassistenten, erlaubt die automatisierte Generierung von:

- Datenmodellen
- Services
- UI-Komponenten
- Tests

Projekte werden im sogenannten *Storyboard* strukturiert und können direkt auf der SAP BTP bereitgestellt werden. Joule unterstützt mit Befehlen eine schnelle und individuelle Entwicklung. Die Entwicklungsumgebung ermöglicht:

- Full-Stack-Entwicklung
- Frontend-Entwicklung
- Backend-Entwicklung
- mobile Entwicklung

Ein zentrales Element von SAP Build ist das Application Lifecycle Management. Es beinhaltet SAP Cloud Transport Management, das die zentrale Steuerung und Protokollierung von Transporten innerhalb der SAP BTP ermöglicht. Es unterstützt sowohl Pro-Code-Entwicklungen wie CAP und ABAP RESTful Application Programming Model als auch Entwicklungen mit SAP Build. Komponenten aus SAP-Build-Apps lassen sich transportieren, indem Projekte exportiert und einem Transportpaket im SAP Cloud Transport Management übergeben werden. SAP Build Process Automation verfügt über eine eigene Funktion namens Promoting Projects, die den Transport von Automatisierungen inklusive ihrer Komponenten erlaubt. Ein zentrales Deployment-Interface kann direkt mit SAP Cloud Transport Management verbunden werden, um Transport und Nachvollziehbarkeit sicherzustellen. Die Nutzung von SAP Build als No-Code- und Low-Code-Tool erfordert ein durchdachtes Governance-Modell. Standardisierte Freigabeprozesse und ein durchgängiges Lifecycle Management sind ebenso wichtig wie technische Werkzeuge zur Umsetzung, etwa SAP Cloud Transport Management. Die IT-Abteilung muss Richtlinien für Prüfungen und Freigaben erstellen, die vor dem produktiven Einsatz von Inhalten eingehalten werden

müssen. Kontinuierliches Monitoring auf Performance und Einhaltung der Richtlinien ist ebenfalls essenziell.

Abbildung 1.17 gibt Ihnen einen Überblick über SAP Build, das in Kapitel 4 genauer betrachtet wird.

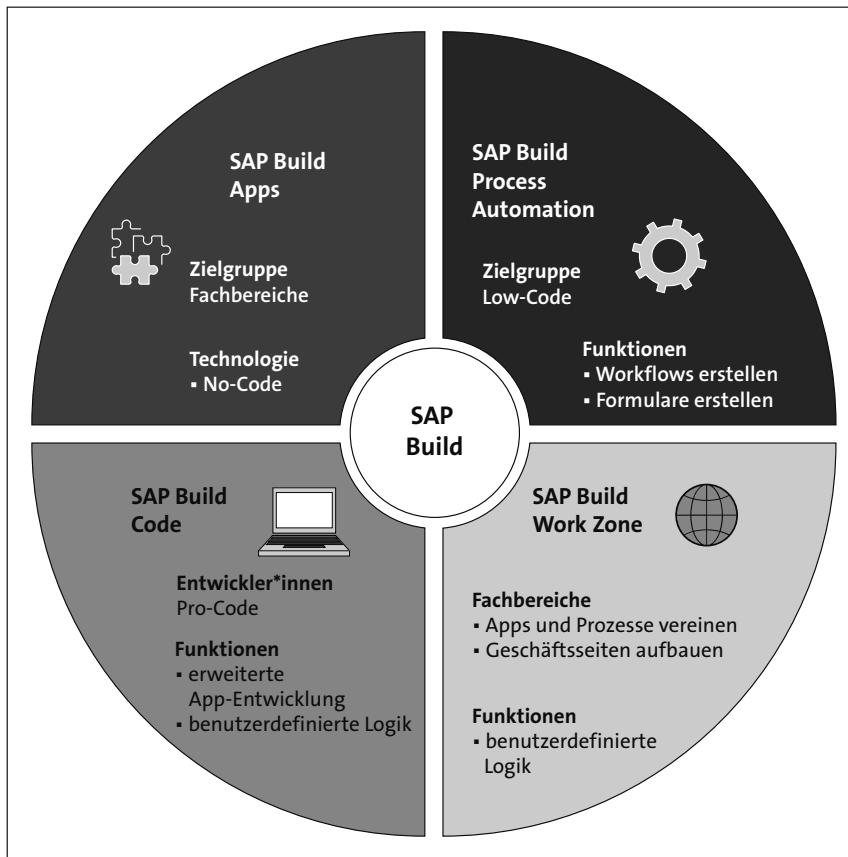


Abbildung 1.17 SAP Build

Ausführliche Informationen zum Thema Anwendungsentwicklung finden Sie in Kapitel 2, »SAP Cloud Application Programming Model«, Kapitel 3, »ABAP in der Cloud und ABAP RESTful Application Programming Model«, und Kapitel 4, »SAP Build«.

## 1.4 Event- und API-gesteuerte Kommunikation

Events und APIs sind die Grundlage der modernen Kommunikation innerhalb der Systemlandschaften. Ereignisse (engl. Events) werden als Auslöser für Informationsflüsse, die Zustandsänderungen oder Aktionen genutzt. Im SAP-System könnte dies

z. B. die Anlage oder Änderung eines Kundenauftrags sein. In einer ereignisgesteuerten Architektur werden solche Events von einem Sender erzeugt und an einen Empfänger übertragen. Die event-gesteuerte Architektur ist ein Software-Designmuster, das auf Ereignissen basiert. Die Übertragung dieser Events erfolgt in den meisten Fällen im Format JSON oder XML. Ein Event Broker Service ermöglicht die strukturierte Verwaltung und Verteilung von Ereignissen. Durch die Nutzung separater Message Clients mit individuellen Konfigurationen lassen sich Verantwortlichkeiten klar zuordnen und Abläufe besser nachvollziehen. Systeme können über verschiedene Programmiermodelle wie CAP oder ABAP RESTful Application Programming Model an die Event-Infrastruktur angebunden werden. SAP Event Mesh ist ein Event Broker Service auf der SAP BTP. SAP Event Mesh besteht aus verschiedenen Message Clients mit eigenen Konfigurationen und Anmeldedaten. Die Nutzung unterschiedlicher Message Clients bietet klare Verantwortlichkeiten und eine verbesserte Nachvollziehbarkeit. Events sind fester Bestandteil des ABAP RESTful Application Programming Models.

APIs spielen eine wichtige Rolle bei der Verbindung von Anwendungen und Diensten. Eine zentrale Plattform oder auch Middleware zur Verwaltung von Schnittstellen ermöglicht die Nutzung, Überwachung und Erweiterung von APIs.

## 1.5 Zusammenfassung

In diesem Kapitel haben Sie die zentralen Grundlagen und strategischen Konzepte der SAP Business Technology Platform kennengelernt. Im Mittelpunkt stand die Clean-Core-Strategie. Sie bildet das Fundament für die Entwicklung cloud-kompatibler Erweiterungen und stellt sicher, dass SAP-Systeme möglichst nah am Standard bleiben.

Wir haben Ihnen die fünf Dimensionen von Clean Core, Extensibility, Data, Integration, Processes und Operations vorgestellt und erläutert, welche Auswirkungen diese auf Entwicklungsprojekte haben und welche Prinzipien dabei zu beachten sind. Besonders wichtig ist die klare Trennung zwischen SAP-Standard und kundenspezifischer Logik.

Ein zentrales Lernziel war die Einordnung der drei Entwicklungsansätze:

- Key-User Extensions
  - ABAP Cloud
  - Side-by-Side Extensions

Diese Ansätze wurden im Three-Tier-Modell systematisch gegenübergestellt, sodass Sie beurteilen können, welche Entwicklungsform für die gewünschten Use Cases am besten geeignet ist. Als Nächstes haben wir Ihnen die Bedeutung der Clean Core Qua-

# Auf einen Blick

1	SAP Business Technology Platform: strategische Grundlagen und Architektur .....	21
2	SAP Cloud Application Programming Model .....	59
3	ABAP in der Cloud und ABAP RESTful Application Programming Model .....	165
4	SAP Build .....	239
5	Events und APIs .....	353
6	Künstliche Intelligenz auf der SAP BTP .....	395
7	DevOps und Sicherheit .....	487

# Inhalt

Vorwort .....	13
Einleitung .....	15

---

## **1 SAP Business Technology Platform: strategische Grundlagen und Architektur** 21

---

<b>1.1 Clean Core .....</b>	22
1.1.1 Grundprinzipien des Clean-Core-Ansatzes .....	23
1.1.2 Technologische Grundlagen: In-App Extensions, ABAP Cloud und Side-by-Side Extensions .....	25
1.1.3 Das Three-Tier-Modell in der Praxis .....	28
1.1.4 Clean Core Quality Gates .....	30
1.1.5 Tools und Services .....	32
<b>1.2 Entwicklungsmöglichkeiten .....</b>	34
<b>1.3 Fünf Säulen der SAP BTP .....</b>	37
1.3.1 Integration .....	38
1.3.2 Daten und Analytics .....	42
1.3.3 Künstliche Intelligenz .....	44
1.3.4 Automatisierung .....	47
1.3.5 Anwendungsentwicklung .....	47
<b>1.4 Event- und API-gesteuerte Kommunikation .....</b>	55
<b>1.5 Zusammenfassung .....</b>	56

---

## **2 SAP Cloud Application Programming Model** 59

---

<b>2.1 Grundlagen vom SAP Cloud Application Programming Model .....</b>	60
2.1.1 Beschleunigter Entwicklungsprozess mit CAP .....	60
2.1.2 Kernkomponenten des SAP Cloud Application Programming Models .....	62
2.1.3 Laufzeitumgebungen .....	64
2.1.4 Domain-Driven Design .....	65
2.1.5 Funktionsumfang .....	67

2.1.6	Einsatzszenarien .....	68
2.1.7	Core Data Services im SAP Cloud Application Programming Model .....	69
<b>2.2</b>	<b>Entwicklung mit dem SAP Cloud Application Programming Model .....</b>	<b>75</b>
2.2.1	Erstellung eines neuen Projekts .....	75
2.2.2	Datenbanken modellieren .....	82
2.2.3	Services entwickeln .....	100
2.2.4	Benutzeroberflächen erstellen .....	136
<b>2.3</b>	<b>Deployment-Strategien .....</b>	<b>150</b>
<b>2.4</b>	<b>Entwicklungstools und -umgebungen für CAP-Projekte .....</b>	<b>158</b>
<b>2.5</b>	<b>Zusammenfassung .....</b>	<b>162</b>

### **3 ABAP in der Cloud und ABAP RESTful Application Programming Model**

---

165

<b>3.1</b>	<b>Cloud-basierte ABAP-Entwicklung .....</b>	<b>166</b>
3.1.1	ABAP Environment .....	166
3.1.2	ABAP Cloud auf SAP S/4HANA .....	173
<b>3.2</b>	<b>Core Data Services im ABAP RESTful Application Programming Model .....</b>	<b>175</b>
3.2.1	Core Data Services in ABAP und das virtuelle Datenmodell .....	175
3.2.2	Evolution von Entwicklungsmodellen in ABAP .....	181
3.2.3	ABAP RESTful Application Programming Model .....	182
<b>3.3</b>	<b>Anwendungen entwickeln .....</b>	<b>217</b>
3.3.1	Datenbanken modellieren .....	217
3.3.2	Services entwickeln .....	221
3.3.3	Benutzeroberflächen entwickeln .....	229
<b>3.4</b>	<b>Anwendungen deployen .....</b>	<b>235</b>
<b>3.5</b>	<b>Entwicklungstools und -umgebungen .....</b>	<b>235</b>
3.5.1	Eclipse .....	235
3.5.2	Business Object Generator .....	236
<b>3.6</b>	<b>Zusammenfassung .....</b>	<b>238</b>

<b>4 SAP Build</b>	239
<b>4.1 No-, Low- und Pro-Code im Vergleich</b>	240
<b>4.2 SAP-Build-Produktporfolio</b>	242
<b>4.3 SAP Build Apps</b>	247
4.3.1 Mit SAP Build Apps entwickeln	247
4.3.2 Web- und mobile Anwendungen	250
4.3.3 Testen, Erstellen und Bereitstellen der Anwendung	277
<b>4.4 SAP Build Process Automation</b>	280
4.4.1 Komponenten	282
4.4.2 Entwicklungsansätze	284
4.4.3 Mit SAP Build Process Automation entwickeln	286
<b>4.5 SAP Build Work Zone</b>	318
4.5.1 Unterschiede zwischen Standard Edition und Advanced Edition	318
4.5.2 Themes	318
4.5.3 Architektur	319
4.5.4 SAP Build Work Zone, Standard Edition	320
4.5.5 SAP Build Work Zone, Advanced Edition	326
<b>4.6 SAP Build Code</b>	341
4.6.1 Befehle in Joule mit SAP Build Code	343
4.6.2 Beispiel für Joule in der Entwicklung einer CAP-Anwendung	345
<b>4.7 Application Lifecycle Management</b>	350
<b>4.8 Zusammenfassung</b>	351
<b>5 Events und APIs</b>	353
<b>5.1 Was versteht man unter Events?</b>	354
<b>5.2 SAP Event Mesh auf der SAP BTP</b>	355
5.2.1 SAP Event Mesh einrichten	356
5.2.2 Message Client erstellen	357

<b>5.3</b>	<b>Systeme an SAP Event Mesh anbinden</b>	360
5.3.1	Anbindung mit dem ABAP RESTful Application Programming Model	360
5.3.2	Anbindung an das SAP Cloud Application Programming Model	364
5.3.3	Kombination von ABAP RESTful Application Programming Model und SAP Cloud Application Programming Model	366
<b>5.4</b>	<b>API Management und Integration</b>	389
5.4.1	SAP Business Accelerator Hub	389
5.4.2	Integration von Drittanbieter-APIs	392
<b>5.5</b>	<b>Zusammenfassung</b>	394

---

## 6 Künstliche Intelligenz auf der SAP BTP 395

<b>6.1</b>	<b>KI-Services und -Werkzeuge auf der SAP BTP</b>	398
6.1.1	Joule	398
6.1.2	SAP AI Services	401
6.1.3	SAP AI Core	405
6.1.4	SAP AI Launchpad	408
<b>6.2</b>	<b>Anbindung von SAP AI Core und SAP AI Launchpad</b>	415
6.2.1	Services registrieren	415
6.2.2	SAP AI Core verbinden	417
6.2.3	Konfiguration anlegen	419
6.2.4	Deployment anlegen	421
6.2.5	Prompt-Editor	422
6.2.6	Prompt mit REST API	424
6.2.7	Orchestrierung	425
<b>6.3</b>	<b>Integration von KI in der Entwicklung</b>	431
6.3.1	SAP Cloud Application Programming Model	432
6.3.2	ABAP in der Cloud	446
6.3.3	SAP Build	466
<b>6.4</b>	<b>SAP HANA Cloud und KI</b>	469
6.4.1	SAP HANA Cloud Vector Engine	470
6.4.2	Machine-Learning-Bibliotheken	475
6.4.3	SAP Knowledge Graph	481
<b>6.5</b>	<b>Zusammenfassung</b>	484

<b>7 DevOps und Sicherheit</b>	487
<b>7.1 DevOps-Prinzipien in der SAP-Entwicklung</b> .....	488
<b>7.2 Versionsverwaltung mit Git</b> .....	490
7.2.1 Befehle in Git .....	491
7.2.2 Branching-Modelle .....	492
7.2.3 Mergen .....	496
<b>7.3 Transportmanagement mit CI/CD</b> .....	497
7.3.1 Transportmanagement mit SAP Cloud Application Programming Model .....	504
7.3.2 Transportmanagement mit ABAP in der Cloud .....	507
<b>7.4 Fehlerbehandlung</b> .....	516
7.4.1 SAP Alert Notification .....	516
7.4.2 Logging .....	522
<b>7.5 Zusammenfassung</b> .....	525
 Autorenteam .....	527
Index .....	529

# Anwendungsentwicklung in der Cloud

## Grundlagen für die Cloud-Entwicklung

Zukunftssicher entwickeln auf der SAP BTP! Erfahren Sie, welche Konzepte hinter der Plattform stecken und wie Sie die verschiedenen Programmiermodelle, APIs und Event-Mechanismen optimal kombinieren.

## Schritt für Schritt zur eigenen Anwendung

Ob Sie mit dem SAP Cloud Application Programming Model, ABAP Cloud oder SAP Build arbeiten: Dieses Buch führt Sie in detaillierten Anleitungen durch den Entwicklungsprozess – von Events und APIs bis zum Deployment.

## KI und DevOps erfolgreich einsetzen

App-Entwicklung mit künstlicher Intelligenz oder Prozessoptimierung mit DevOps: Die Autoren zeigen Ihnen, wie Sie neue Technologien und Methoden gewinnbringend für Ihre Projekte einsetzen.

»Dieses Buch führt Sie systematisch in die Welt der SAP BTP ein.«

## Auf einen Blick

- Event- und API-gesteuerte Kommunikation
- SAP Cloud Application Programming Model
- Core Data Services
- ABAP Cloud
- ABAP RESTful Application Programming Model
- Low-Code-Entwicklung mit SAP Build
- Events und APIs
- Anbindung von KI-Services
- Joule
- DevOps und Sicherheit



### Das Autorenteam

Florian Pohl, Jean-Francois Tate Tsiledze, Constantin Freiherr Teuffel von Birkensee und Raphael Witte sind Experten bei der NTT DATA Business Solutions AG mit umfassender Erfahrung in modernen SAP-Technologien und der Cloud-Entwicklung. Sie unterstützen Kunden bei Projekten in den Bereichen Prozessautomatisierung, KI, Side-by-Side-Entwicklung und SAP-BTP-Architektur.

