

# Monolith vs. Composable:

Wie eine zukunftsfähige IT-Infrastruktur für eine herausragende Customer Experience gelingt.

Argumente, Beispiele und eine Checkliste für Unternehmen, die neue, flexible Softwarelösungen suchen.



# Inhalt

Einleitung	3
Vorteile einer Composable-CX-Plattform	7
Herausforderungen einer Composable-CX-Plattform	10
Unterstützung bei der Realisierung	12
Auswahl der passenden CX-Plattformstruktur	14
Checkliste zur Systemwahl	17
Resümee	18



# Einleitung

## Status Quo der Unternehmens-IT

Viele Unternehmen sind mit der Leistungsfähigkeit ihrer Customer-Experience-Systemlandschaft unzufrieden. Laut Gartner glauben 54 %, dass die IT des Unternehmens die Customer Experience ausbremst. Häufig sind monolithische Systeme im Einsatz, die mit zahlreichen Anpassungen und Eigenentwicklungen zunehmend komplex geworden sind. Zudem sind die Systeme inzwischen in die Jahre gekommen.

Fehlende Performance, mangelhafte Skalierbarkeit sowie langwierige Anpassungsprojekte sind deshalb die häufigsten Gründe, um nach anderen Lösungen Ausschau zu halten. Der Wunsch ist dabei, nicht gleich alles über Bord zu werfen, sondern nur die Komponenten zu ersetzen, die die Entwicklung besonders hemmen. ***Denn wer kauft schon ein neues Fahrrad, nur weil es einen Platten hat?***

“ ***Denn wer kauft schon ein neues Fahrrad, nur weil es einen Platten hat?*** ”





54%

glauben, dass die IT des Unternehmens die Entwicklung der Customer Experience ausbremst.<sup>1</sup>

## HERAUSFORDERUNGEN BEI DER UMSETZUNG DER CX-STRATEGIE<sup>2</sup>

Basis: 176 Unternehmen | Filterung: Kanalübergreifende CX-Strategie umgesetzt, in Erarbeitung oder in Planung | Mehrfachnennungen



<sup>1</sup> 2022 Gartner CIO and Technology Executive Survey

<sup>2</sup> 2CX Trends 2024 – eine Studie von technoconsult GmbH mit Unterstützung von KPS

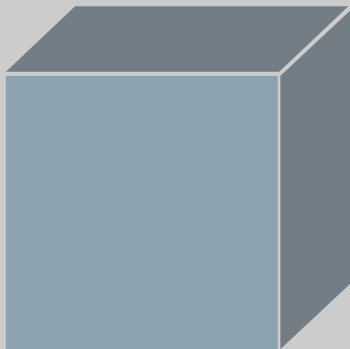
## Ein System oder viele Module – die Unterschiede kurz erklärt

Die Frage lässt sich mit einem klaren “Es kommt darauf an” beantworten. Je nach Unternehmen kann ein Monolith mit allen wichtigen Funktionen die beste Lösung sein, während andere Unternehmen möglicherweise mehr Agilität benötigen. Schauen wir aber zunächst, was die beiden Ansätze ausmacht:

Auf der einen Seite steht ein **Monolithisches System**, das in der Regel von einer Softwarefirma stammt und das man sich als starren Stein vorstellen kann. Dieses erfüllt meist alle grundlegenden Anforderungen, bietet eine zentrale Datenhaltung und hat sich über viele Jahre bewährt. Für Sonderfunktionen muss es jedoch aufwendig und kostspielig angepasst werden.

Dem gegenüber steht eine Zusammenstellung von modularen Einzelsystemen, die miteinander interagieren und im Ergebnis deutlich mehr als nur die grundlegenden Anforderungen abdecken. Der Austausch von Kundendaten und -interaktionen erfolgt dabei über APIs, kundenspezifische Prozesse werden in externe Microservices ausgelagert – und zwar ohne eines der Module grundlegend in seiner Struktur “zu verbiegen”. Dies ist als **Composable Architecture** bekannt.

Ein monolithisches System,  
das unterschiedliche Aufgaben  
moderat erfüllt



Multiple, verbundene Komponenten,  
die jede spezifische Aufgabe  
perfekt erfüllen



**In der Realität findet sich häufig eine Mischform aus diesen beiden Ansätzen.**

Typische Einzelbereiche der Customer Experience, die durch diese Architektur abgebildet werden, sind:

- ✓ Marketing
- ✓ Sales
- ✓ Kundenservice
- ✓ Kundendatenmanagement
- ✓ Loyalty
- ✓ eCommerce
- ✓ Search
- ✓ Marktplätze
- ✓ Analytics & AI
- ✓ Content & Assets
- ✓ Personalisierung



# Implementierung vs. Orchestrierung

Unterschiede der beiden Systemansätze zeigen sich auch in der Einführung der Lösungen.

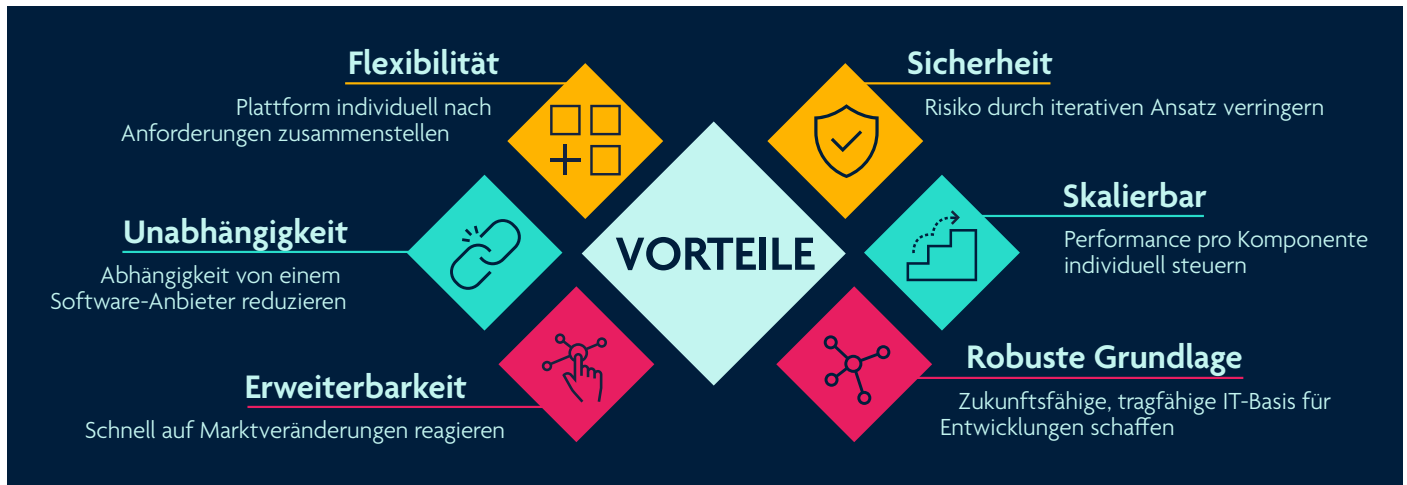
**Ältere und eher zentralisierte Systeme** (u.a. auch Smart-Hubs) zeichnen sich dadurch aus, dass sie eine Vielzahl an Funktionen abdecken und für unternehmensspezifische Anforderungen angepasst werden können. Vor dem Live-Gang steht also eine mehr oder minder große Anpassungs- und Implementierungsphase. Kennzeichnend dafür ist, dass Entwickler die Software an die Prozesse und Workflows des Unternehmens adaptieren. Dies hat weiter unten ausgeführte Vorteile für das Unternehmen, aber Nachteile bei Software-Upgrades und wenn das Gesamtgebilde flexibel erweitert werden soll.

Dem gegenüber steht die **moderne Composable Architecture**, bei der das Gesamtsystem aus einzelnen Komponenten zusammengesetzt ist. Die Komponenten arbeiten oft Cloud- und API-basiert und geben Standards und Workflows vor. Die gewünschte Funktionalität ergibt sich aus der Auswahl der am besten passenden Komponenten. Eine Anpassung an Unternehmensworkflows erfolgt selten; hier muss sich das Unternehmen an die branchenüblichen Standards anpassen. APIs und Microservices in der Cloud sorgen für die Orchestrierung des Gesamtsystems (MACH) und bieten Möglichkeiten, kundenspezifische Prozesse, die nicht durch die Komponenten abgebildet werden können, zu erweitern. Damit kann für jede Anwendung ein anderer Anbieter zum Einsatz kommen und somit eine optimale Zusammenstellung erzielt werden.



# Vorteile einer Composable-CX-Plattform

## Überblick



## Individuell und unabhängig

Wer seine IT-Systeme nicht aus einer Hand beziehen muss, kann eine Best-of-Breed-Strategie verfolgen und sich die jeweils besten am Markt erhältlichen Systeme herauspicken. Die Auswahl erfolgt nicht danach, was der eine Plattform-Anbieter zu bieten hat, sondern was am besten zu den eigenen Unternehmensanforderungen passt. Dies gilt auch dann noch, wenn sich die Bedürfnisse ändern und man zu einer anderen Lösung wechseln möchte. Denn auch das ist einfach möglich.

Damit vermeidet die IT bei der Systemwahl ein Vendor-Lock-In und den oft damit einhergehenden Zwängen: Das eigene Unternehmen muss keine überzogenen Preise oder geringere Funktionalität akzeptieren. Diese Unabhängigkeit bringt das Unternehmen auch in eine bessere Verhandlungsposition, weil es nur noch losere Bindungen an mehrere Anbieter gibt und sich Partner einfacher wechseln lassen, wenn es opportun erscheint.



## Erweiterbar und hochverfügbar

Der rasante Aufstieg von Künstlicher Intelligenz zeigt, wie wichtig es ist, schnell reagieren und seine CX-Landschaft anpassen zu können. Kundinnen und Kunden erwarten schon jetzt einen intelligenten Service-Bot und Kaufberater auf der Website.

Eine Composable Architecture macht es leicht, Komponenten auszutauschen und neue Funktionalität hinzuzufügen. Dafür lassen sich weitere Online-Dienste anbinden oder größere Komponenten austauschen. Auch größere Prozess-Anpassungen lassen sich in einer Composable Architecture leichter implementieren, als dies in der Vergangenheit der Fall war.

Darüber hinaus sind Systemänderungen kein Grund für Down-Times der gesamten Plattform. Wird nur das Loyalty-Modul mit einem Update versehen, läuft der Rest des Systems weiter. Updates müssen daher nicht nachts oder am Wochenende stattfinden, sondern lassen sich zu normalen Zeiten ohne Beeinträchtigung anderer IT-Komponenten durchführen. Das ist insbesondere wichtig, wenn hochfrequentierte Kassensysteme im Einsatz sind und 24/7 bereitstehen müssen.







## Performant und robust

Muss die Customer-Experience-Plattform eine Vielzahl von Verkaufstransaktionen pro Sekunde verarbeiten können, spielt die Verarbeitungsperformance eine große Rolle. In einer Composable-CX-Plattform kann man gezielt für einzelne Komponenten mehr Systemressourcen allokalieren, was zu einer höheren Performance führt und das Verarbeiten von großen Datenmengen ermöglicht.

Das erleichtert die Skalierung bei Peak-Events wie einem Black Friday oder Werbekampagnen. Es vermeidet außerdem das pauschale Oversizing von IT-Systemen, nur weil wenige Komponenten auf große Datenvolumina in Echtzeit reagieren müssen.

Eine Composable Architecture bildet nicht zuletzt eine zukunftsfähige Grundlage, die robust und stabil funktioniert und sich jederzeit an neue Gegebenheiten anpassen lässt. Die Grundstruktur ist dabei so gewählt, dass der Austausch zahlreicher Komponenten die Gesamtstabilität nicht beeinträchtigt und damit Anpassungen leicht macht. So verfügen Unternehmen immer über eine topmoderne Best-of-Breed-Lösung, die eine sichere Basis für zukünftige Anforderungen bildet und sich nach den Bedürfnissen des Unternehmens richtet.



# Herausforderungen einer Composable-CX-Plattform



Neben den vielen Vorteilen bringt die Composable Architecture auch einige Herausforderungen mit sich.

## Technische Komplexität

Mit dem Einsatz von vernetzten Anwendungen verschiedenen Anbieter steigen die technische Komplexität und der Administrationsaufwand. Sind alle relevanten Daten über verschiedene Systeme verteilt, muss exakt festgelegt sein, wie die Komponenten Daten verarbeiten, synchronisieren und abfragen. Es gilt nicht mehr nur ein zentrales System zu betreuen, sondern mehrere verschiedene, die alle unterschiedliche User-Interfaces bereitstellen. Das kann auch zum Auseinanderbrechen einheitlicher Prozesse und unterschiedlichen

Entwicklungen in einzelnen Abteilungen führen. Hier gilt es, durch entsprechende Strukturen ein einheitliches Vorgehen sicherzustellen.

Für die IT-Abteilung verschieben sich außerdem die Tätigkeitsfelder. Statt der Implementierung innerhalb einer genau festgelegten Umgebung wie einem eher monolithischen SAP-CRM-System, müssen auf Cloud-Systemen headless Micro-Services und APIs konfiguriert und programmiert werden (MACH).

## Die MACH-Architektur kurz erklärt:

**MACH** steht für Microservices, API-First, Cloud-Native und Headless:

**Microservices-orientiert:** Die Architektur betont die Verwendung von Microservices, um eine Anwendung in kleine, unabhängige Dienste aufzuteilen, die über APIs kommunizieren.

**API-First:** Der Fokus liegt auf der klaren Definition und Nutzung von APIs, um die Kommunikation zwischen den verschiedenen Komponenten der Anwendung zu ermöglichen. Dies erleichtert die Skalierbarkeit und den Austausch von Diensten.

**Cloud-Native:** MACH-Anwendungen sind darauf ausgerichtet, Cloud-Technologien und -Infrastrukturen zu nutzen, um Skalierbarkeit, Flexibilität und Effizienz zu verbessern.

**Headless:** In der Headless-Architektur ist die Benutzeroberfläche (Frontend) von den Backend-Services entkoppelt. Dies ermöglicht es, unterschiedliche Frontend-Technologien zu verwenden und die Benutzeroberfläche flexibel zu gestalten.

# Effektives Kostenmanagement

Beim Wechsel zu einer Composable Architecture verändern sich auch das Kostenmanagement, die Lizenzierung und die Budget-Planung. Statt nur mit einem Partner zu verhandeln, können es schnell viele verschiedene Dienstleister sein. Oft werden die IT-Kosten auch im eigenen Unternehmen auf verschiedene Budgets aufgeteilt. Damit verteilen sich eventuell die Verantwortlichkeiten, die vorher nur in einer Hand waren, auf mehrere Personen.

Unternehmen sollten diese Veränderung nicht unterschätzen. Es ist deshalb wichtig, Transparenz herzustellen. Nur dann lässt es sich vermeiden, dass die Einzelbudgets aus dem Ruder laufen. Natürlich ist es auch möglich, Kosten einzusparen, weil nur die Funktionalität gebucht wird, die die Fachabteilungen benötigen. Sind Budget- und Entscheidungsbefugnisse dazu nicht mehr in einer Hand, sind dafür regelmäßige Abstimmungsrunden notwendig.

Beim Einsatz einer Composable-CX-Plattform kann es auch nötig sein, ein professionelles Lizenzmanagement einzuführen.

Da diese Herausforderungen gut gemeistert werden wollen, lohnt es sich in der Regel, mit einem erfahrenen Partner zusammenzuarbeiten. Mit zielgerichteter Unterstützung bei der Orchestrierung/der Integration der Komponenten lässt sich eine Composable-CX-Plattform, die alle Unternehmensanforderungen optimal erfüllt, zielgerichtet umsetzen. Worauf kommt es aber bei der Auswahl eines Partners für ein solches Projekt an? Im Folgenden gehen wir auf die nötige Expertise und Erfahrung ein, die ein Partner mitbringen sollte.





# Unterstützung bei der Realisierung

## Anforderungen an Projektpartner



Bei der Auswahl des Partners für die Konzeption und Implementierung einer Composable Architecture gibt es Besonderheiten zu berücksichtigen. Da in diesem Fall nicht ein System alle Funktionen bereitstellt, braucht es Partner mit einer breiten Marktkenntnis und technischer Expertise. Er muss wissen, in welchen Cloud-Anwendungen und Microservices was abgebildet und wie diese passend kombiniert und orchestriert werden können. Dazu sollte der Partner auf zahlreiche Best Practices zurückgreifen können und entsprechende Projekte erfolgreich gestemmt haben.

Viele Beratungsunternehmen haben sich außerdem häufig spezialisiert. Sie verfügen entweder über detaillierte Kenntnisse im ERP-Umfeld und bei der Implementierung von Back-End-Lösungen. Oder sie sind besonders gut aufgestellt bei der Entwicklung von Front-Ends und modernen CX-Lösungen. Bei einer Composable Architecture ist aber beides gefragt. Der Partner sollte daher sowohl im Back-End- wie im Front-End-Bereich große Expertise besitzen.

# Den richtigen Partner auswählen

Der Aufbau einer Composable Architecture erfordert also mehr Fähigkeiten als die Implementierung und Anpassung einer Softwarelandschaft:

**Umfassende Orchestrierung:** Die Fähigkeit, aus der Vielzahl der Systeme mit minimalem Aufwand ein funktionierendes, robustes, sicheres Gesamtsystem zu konzipieren, erstellen und implementieren.

**Erfahrung:** Umfassende Projekterfahrung und Branchenexpertise bei der Transformation von On-Premises-Prozessen in die Cloud.

**Technologische Expertise:** Das Wissen, welche Standard-Komponenten und Tools am Markt angeboten werden und welche reibungslos zusammenarbeiten, um die Anforderungen am besten zu erfüllen.

**Business Change:** Die Anwender darin unterstützen, die Potenziale der verschiedenen Komponente zu erkennen und zu nutzen.

**Front- & Back-End-Exzellenz:** Eine umfassende Omnichannel-Erfahrung für die nahtlose Implementierung von CX-Prozessen und Back-End-Funktionalitäten.





# Auswahl der passenden CX-Plattformstruktur

## Für wen eignet sich eine **monolithische Architektur**?

Nicht jede Lösungsarchitektur ist für jedes Unternehmen geeignet. Sind die IT-Abteilung oder das IT-Budget stark limitiert, ist ein eher monolithischer Smart-Hub-Ansatz, der out of the box schon die meisten Funktionen bereitstellt und schnell einsatzbereit ist, ein guter Startpunkt. Smart-Hubs lassen sich außerdem häufig um Composable-Elemente erweitern. Damit sind die Initialaufwände und Anfangskosten gering und es steht schnell ein robustes System zur Verfügung.

Ist das Unternehmen zwar klein, weiß aber jetzt schon, dass es sich schnell entwickeln und weitere CX-Systeme benötigen wird, ist der anfangs verlockend wirkende, günstige Smart-Hub eventuell die falsche Wahl. Viele Smart-Hub-Systeme sind beispielsweise nicht dafür ausgelegt, in kurzer Zeit große Datenmengen parallel in Echtzeit zu verarbeiten. Insbesondere in Peakzeiten könnte es daher zu Performance-Problemen kommen.

Sind außerdem viele Anpassungen notwendig, weil die eigenen Prozesse nur mit zahlreichen Anpassungen abbildbar sind, ist der Kostenvorteil schnell dahin, weil initial hohe Entwicklungskosten und lange Projektlaufzeiten entstehen.



## Für wen eignet sich eine **Composable Architecture**?

Je größer der Wunsch nach flexiblen, hoch performanten Einzelkomponenten ist, um so eher eignet sich eine Composable Architecture als Lösungsansatz. Sie kommt besser mit häufigen Änderungen und wachsenden oder schwankenden Leistungsanforderungen zurecht. Das Ergebnis einer solchen Konzeption lässt sich exakt auf die Bedürfnisse eines Unternehmens abstimmen.

Allerdings kostet das auch: im wörtlichen und übertragenen Sinn. Einerseits erfordert das Management mehrerer Einzelsysteme im Software-Verbund erfordert mehr IT-Ressourcen, um die einzelnen Komponenten einzurichten und zu betreuen sowie das Gesamtsystem zu orchestrieren. Andererseits kann auch vorkommen, dass die Einzellizenzen insgesamt ein höheres Budget erfordern als ein Einzelsystem.

Unternehmen müssen sich außerdem darüber klar sein, dass Composable-Komponenten nicht an Unternehmensprozesse angepasst, sondern durch Microservices und APIs erweitert werden. Das erfordert, dass sich Workflows an die Standard-Prozesse der Komponenten anpassen müssen - und nicht umgekehrt die Software an die über Jahre gewachsenen Prozessstrukturen des Unternehmens. Die Flexibilität des Systems hat also den Pferdefuß, dass sich Prozesse und Workflows im Unternehmen ändern werden und das organisatorisch abgebildet und durch Weiterbildungen unterstützt werden muss.



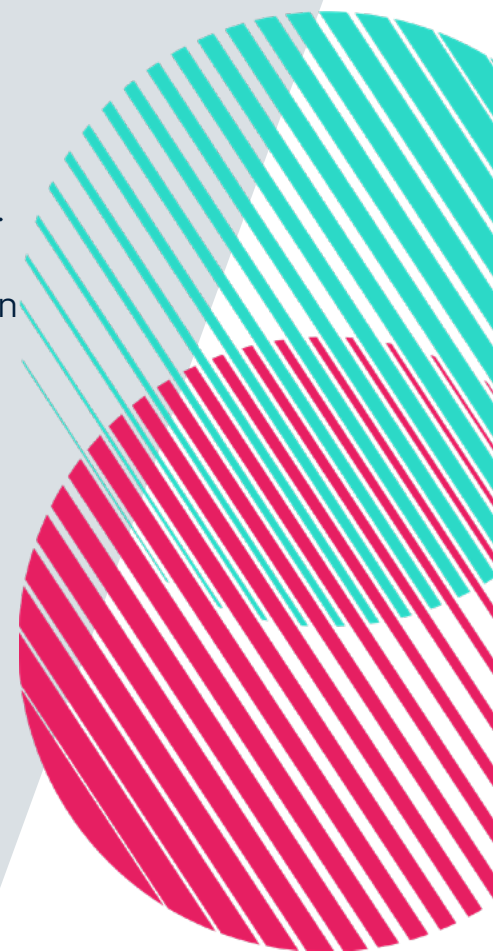
Eine Composable-CX-Plattform kann zunächst also durchaus Mehraufwand verursachen. Dieser macht sich durch eine höhere Performance und Stabilität aber bezahlt: Manche Unternehmen erzeugen über große Webauftritte und viele Filialen (mit mehreren Kassensystemen) ein enormes Datenvolumen, das in Echtzeit verarbeitet werden muss. Diese Systeme müssen nicht nur eine Zahlung abwickeln, sondern auch noch Loyalty-Systeme abfragen und bedienen und Bestätigungsmails verschicken. Bei mehreren hundert oder tausend Vorgängen pro Sekunde, müssen die Back-End-Systeme entsprechend ausgelegt sein. Gerade Kassen- oder Loyalty-Systeme dürfen auf keinen Fall ausfallen und müssen auch zu Peakzeiten zuverlässig funktionieren. Das gelingt häufig nur mit einer Composable-Architecture.

In einer monolithischen Architektur führen Updates aber zu Downtimes des Gesamtsystems. Das macht ein Update organisatorisch aufwendig - entsprechend vermeiden Unternehmen häufige Anpassungen. Das ist ein Nachteil zur Konkurrenz, wenn diese schneller modernisieren kann. Composable-CX-Systeme sind hier im Vorteil, weil sich Teile unabhängig voneinander aktualisieren und erweitern lassen. Die Unabhängigkeit der Einzelkomponenten macht das Gesamtsystem sehr robust gegen Umsatzeinbußen durch Systemanpassungen.

## Wann ist eine **Mischung** aus beiden Ansätzen sinnvoll?

Die Vorteile einer Composable Architecture sind also vielfältig, wie wir gesehen haben. Bei der Einführung einer solchen Architektur befürchten viele jedoch ein aufwendiges IT-Projekt. Dabei muss das gar nicht sein. Über die Jahre haben sich häufig bewährte Prozesse und Funktionalitäten in einer monolithischen Systemstruktur entwickelt. Die muss man nicht gleich über Bord werfen. Sie können oftmals um einzelne Komponenten strategisch erweitert und damit an veränderte Anforderungen angepasst werden.

Auf diese Weise kann man seine Systemlandschaft Schritt für Schritt in die Zukunft führen und robuster, flexibler und performanter gestalten. Auf dem Weg dorthin kann man heute außerdem von Architektur-Templates profitieren, die schon einzelne Module vordefiniert haben und die schnell einsatzbereit sind. So braucht man nicht bei null anzufangen, sondern kann einen einfacheren Einstieg in die Composable Welt finden.



# Checkliste zur Systemwahl

## Ermitteln Sie, was Ihr Unternehmen benötigt

Die folgenden Fragen helfen Ihnen, sich darüber klar zu werden, ob sie eine Composable-CX-Architecture benötigen oder auch mit einem eher monolithischen System (z.B. einem Smart-Hub) zurechtkämen.

Ist es Ihnen wichtig, in allen Bereichen "Best-of-Breed"-Lösungen einzusetzen?	JA	NEIN
Sind Sie ein innovatives und schnell wachsendes Unternehmen?	JA	NEIN
Müssen Sie häufig Anpassungen an Ihrer Systemlandschaft vornehmen?	JA	NEIN
Wenden Sie hohe Kosten und viel Zeit für Software-Anpassungen auf?	JA	NEIN
Ist Ihr Konkurrenzdruck, auf Marktveränderungen schnell zu reagieren, hoch?	JA	NEIN
Gibt es Daten, die hoch performant, großvolumig oder in Echtzeit verarbeitet werden müssen?	JA	NEIN
Haben Sie IT-Kapazitäten, komplexere Systemlandschaften zu orchestrieren und mehrere Anbieter zu managen?	JA	NEIN
Lassen Ihre IT-Kapazitäten einen Wechsel auf eine andere Architektur zu?	JA	NEIN
Möchten Sie zukünftig vermehrt auf Standard-Prozesse setzen?	JA	NEIN

Lautet Ihre Antwort größtenteils „Ja“, empfiehlt sich eine modulare Architektur. Lautet sie größtenteils „Nein“, bietet sich ein Smart-Hub mit der Option auf spätere Erweiterungen an.



## Resümee

Monolith, Composable oder ein hybrider Ansatz – auf die eigenen Anforderungen kommt es an. Modulare IT-Systemlandschaften bieten sich besonders an, um auch zu Spitzenzeiten eine optimale Customer Experience zu gewährleisten, was sich positiv auf den Umsatz auswirkt. Eine Composable-CX-Plattform ist besonders robust, performant und flexibel und hat damit den monolithischen Systemen einiges voraus.

Die Einführung einer Composable-CX-Architecture erfordert aber auch ein Umdenken im Unternehmen: Teams müssen sich an etablierte und bewährte Software-Standards anpassen, nicht umgekehrt. Dafür genießt man den Vorteil, einzelne Komponenten schneller austauschen und damit auf Trends reagieren zu können.

Dabei muss die Wahl des passenden Systems keine Entweder-Oder-Frage sein. Denn auch ein hybrider Ansatz ist möglich

und kann die Vorteile von Monolithen mit denen einer Composable Architecture optimal zusammenbringen. Viele Prozesse und Funktionen haben sich häufig über lange Jahre entwickelt und bewährt. Diese muss man nicht ersetzen, sondern kann sie durch einzelne Module erweitern.

Das nötige Know-how für die Erweiterung um einzelne Module oder für die Orchestrierung des Gesamtsystems als Composable-CX-Plattform kann man sich über den richtigen Partner dazuholen. Man muss nur darauf achten, dass dieser über tiefe technische und prozessuale Expertise verfügt, einen Lösungsmarktüberblick hat und umfassende Best Practices aus Projekten mitbringt. Im Idealfall hat der Partner auf Basis dieser Best Practices bereits Architektur-Templates entwickelt, die den Einstieg in die Composable Architecture vereinfachen und beschleunigen, sodass daraus ein überschaubareres Projekt wird.



# Sie wollen Ihre Customer Experience verbessern und den Umsatz steigern? Composable Architecture könnte die Lösung sein?

Sprechen Sie uns an. Wir unterstützen Sie gerne bei der Wahl und Umsetzung der für Ihr Unternehmen besten Zielarchitektur.

✉ [booster@kps.com](mailto:booster@kps.com)



## Interessiert an einem Schnellstart zu mehr Flexibilität und Performance?

Lernen Sie unseren Composable Booster kennen, das Architektur-Template, in dem Commerce, Content Management und weitere CX-Module bereits vordefiniert sind.

**ERFAHREN SIE MEHR**

COMPOSABLE  
**BOOSTER**  
built by KPS

**KPS**