



# Headless CMS im Unternehmen

Was vor dem Start eines Headless CMS-Projekts beachtet werden sollte



The B2B Game Changer

*elbformat*

# Inhaltsverzeichnis

- 03** Einführung: Headless CMS
- 05** Von Killer-Apps zu Killer-APIs
- 07** Nutzeranforderungen und Produkteigenschaften
- 08** Komplexität und Robustheit der Implementierung
- 10** Projekt- und langfristige Wartungskosten
- 11** Berücksichtigung des kontinuierlichen Wandels
- 12** Zusammenfassung

# Headless CMS

## Einführung

Seit mehreren Jahren lässt sich der Trend zum Headless CMS beobachten. Heute ist es bereits eine Grundvoraussetzung für Implementierungen für kleinere Unternehmen und Kampagnen-Webseiten. Auch bei Unternehmensprojekten gewinnt es zunehmend an Bedeutung.

Bevor Sie sich jedoch kopfüber in ein Headless CMS-Projekt stürzen, sollten Sie sich überlegen, wie es zu den Bedürfnissen Ihres Unternehmens passt.

Wie so oft in der IT ist auch der Begriff „Headless CMS“ nicht eindeutig. Daher wollen wir zunächst einen kurzen Überblick über einige der im E-Book verwendeten Begriffe geben:

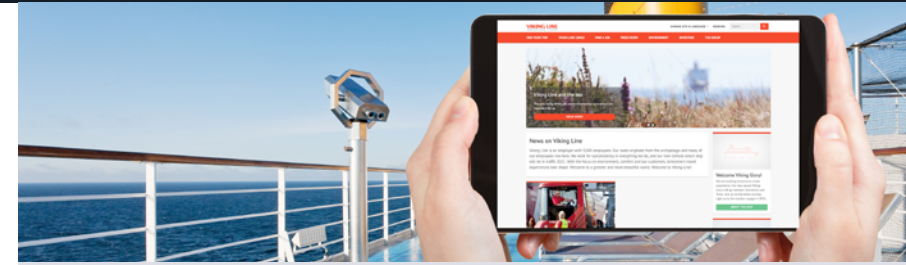
- **Content API:** Tool, das von Redakteuren zur Content-Verwaltung verwendet wird. Liefert einen Roh-Content-Feed zur Nutzung durch andere Entwicklungs-Tools – nicht die Benutzer.
- **Anzeigeschicht:** System, das den von einer Content-API bereitgestellten Feed zur Anzeige für Benutzer rendert, z.B. eine Webseite oder eine mobile App. In diesem E-Book konzentrieren wir uns auf das Web.

Für beide Komponenten gibt es eine Reihe von Optionen auf dem Markt. Um mehr über die verschiedenen Content-API-Optionen zu erfahren, [empfehlen wir unsere Einführung in „Content as a Service“](#).

Für Anzeigeschichten, die von der API bereitgestellte Inhalte nutzen, ist die Systemlandschaft vielfältiger. Ein benutzerdefiniertes Frontend ist ebenfalls möglich. Nebenstehend finden Sie einige bekannte Technologien für Web-Frontends für ein Headless CMS.

Die Kombination aus einer Content-API und einer Anzeigeschicht ergibt ein Headless CMS. Es unterscheidet sich von einem typischen, traditionellen CMS, bei dem Content-API und Anzeige stärker miteinander gekoppelt sind und ein einziges System bilden.

Beide Ansätze haben ihre Vorteile und Herausforderungen, von denen wir einige in diesem E-Book erörtern wollen.



## Technologien für den Aufbau eines Frontends für ein Headless CMS:

### Static Website Generators:

Innerhalb eines Publishing-Prozesses wird eine Reihe von HTML-Dateien erzeugt. Hohe Skalierbarkeit zu niedrigen Kosten. Sehr sicher, kann aber bei dynamischen Funktionen oder dort, wo Berechtigungen auf Benutzerebene für den Zugriff auf Inhalte durchgesetzt werden müssen, einschränkend wirken.

**Beliebte Produkte:** Gatsby.js, Hugo, Jekyll

### Serverseitige JavaScript Frameworks:

Web-Framework, das auf einem Server läuft. Holt Content dynamisch von der Content-API. Hohe Flexibilität für Integrationen, Berechtigungen usw. Erfordert eine Hosting-Umgebung und aktive Wartung.

**Beliebte Produkte:** Express, Next.js, Koa

### Function as a Service (FaaS):

Vereinfachtes Framework, das vollständig verwaltet wird. Preis und Leistungsumfang richten sich nach der Nutzung. Die Landschaft für Headless-CMS-Frontends ist unausgereift und erfordert einige Investitionen in die kundenspezifische Entwicklung. Auch eine Herstellerabhängigkeit ist zu berücksichtigen.

**Beliebte Produkte:** Azure Functions, AWS Lambda, Google Cloud, Open FaaS





# Headless CMS im Unternehmen





# Von Killer-Apps zu Killer-APIs

Content Management Systeme sind heute ausgereift, d.h. es gibt viele Erfahrungswerte und bewährte Verfahren. CMS-Implementierungsprojekte gehören damit zu den eher eintönigen IT-Routinearbeiten.

Die Herausforderungen liegen in der Geschäftslogik, während die Technologie selbst für „Techies“ nicht mehr sehr aufregend sein dürfte. Headless-Content-Management-Tools, die auf der API-Hype Welle reiten, stellen diesen Status quo mit einem alternativen Ansatz für die Content-Bereitstellung in Frage.

APIs (Application Programming Interfaces) gibt es seit den Anfängen der Informatik. Meist handelte es sich um interne APIs, die unmittelbar für die Softwareentwicklung genutzt wurden: Microsoft Windows-Anwendungen interagieren mittels dessen eigener APIs mit der Hardware.

Die neuerliche Aufmerksamkeit gilt vor allem verteilten APIs in Verbindung mit Webtechnologien. Im Headless CMS manifestiert sich dieser Trend für das Content Management.

Aber genau wie eine API in einem Desktop-Betriebssystem ist eine Web-API an sich nur ein Tool. Ihr Wert liegt in den Features und Funktionalitäten, die mit ihr aufgebaut werden können. So konnte Microsoft dank der Killeranwendungen, die Entwickler mit Windows-APIs erstellt haben, die Computerwelt mehr als zwei Jahrzehnte lang dominieren.

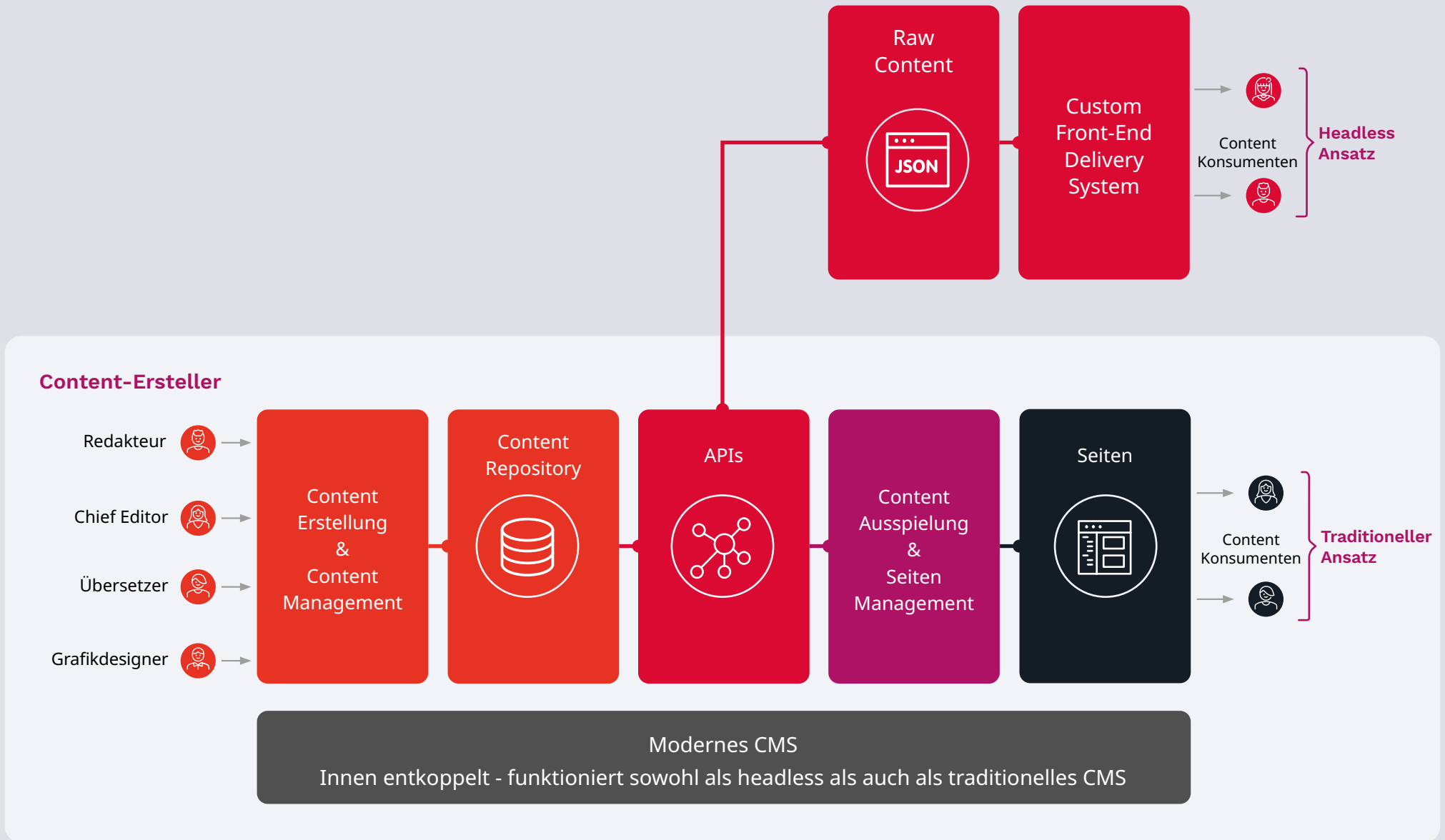
Mit dem Aufkommen von Web und Smartphones verlor das Desktop-Betriebssystem seine dominierende Stellung. Das Unternehmen musste das akzeptieren und sich von der Abhängigkeit von Windows verabschieden, um in der neuen Welt relevant zu bleiben: Es führt Cloud-Dienste und abonnementbasierte Preise für seine Office-Suite ein.

Aber die Dinge sind selten monoton. Desktop-Computing ist sicherlich keine Wachstumsbranche mehr. Dennoch gibt es Bereiche, in denen es nach wie vor die beste Option ist – von Anwendungen, in denen absolute Echtzeit-Präzision erforderlich ist, z.B. in der Musikproduktion, bis hin zu alltäglichen

Tätigkeiten, wie der Erstellung von Blog-Beiträgen und der Erledigung von gelegentlichen Aufgaben. Der PC-Besitz hat seinen Zenit erreicht und ist im Rückgang begriffen. Dennoch arbeitet der Großteil im Büro weiterhin jeden Tag mit einem Desktop- oder Laptop-Computer.

Im Content Management Bereich ist das ähnlich. Für etablierte CMS-Anbieter ist es leicht, auf die Mängel reiner Headless-Systeme hinzuweisen. Auf der Gegenseite produzieren die Marketing-Teams der CaaS-Anbieter fleißig Inhalte, die auf die Unzulänglichkeiten der „alten“ CMS-Produkte hinweisen. Dabei gibt es je nach Anforderungen und Anwendungsfällen durchaus Raum für beide Ansätze.

Wir bei Ibexa sehen keine Notwendigkeit für eine Gegenüberstellung. Dank unserer Produktinfrastruktur können wir interne APIs erstellen und dann über Web-APIs wie REST oder GraphQL verfügbar machen. Sie können die Ibexa DXP als „headless“, „full stack“ oder etwas dazwischen einsetzen. Sie haben die Wahl.



# Nutzeranforderungen und Produkteigenschaften

## Ein Headless CMS dient in erster Linie der Content-Verwaltung

Der Begriff CMS beschreibt heute ein System, dessen Funktion weit über das Erstellen, Modifizieren und Editieren von Content hinausgeht.

Diese Systeme sind über ihren ursprünglichen Umfang längst hinausgewachsen und befinden sich auf dem Weg zu einer DXP (Digital Experience Platform). Roland Benedetti, SVP Strategy von Ibexa, hat dieses Thema in einem [Blogbeitrag mit dem Titel „Was ist eine DXP im Vergleich zu einem CMS“](#) ausführlich erläutert.

Ein Headless CMS ist seiner Definition treuer. Es wird in erster Linie zur Content-Verwaltung eingesetzt. Dadurch ist es für Benutzer leichter zugänglich, „kann“ aber weniger als ein traditionelles Unternehmens-CMS. Diese Ambivalenz kann im Implementierungsprozess für einige Überraschungen sorgen. Einige Features, die den Rahmen eines CMSs sprengen (und schon immer gesprengt haben), sind für Sie vielleicht selbstverständlich.

Wenn Sie ein vorhandenes traditionelles CMS oder DXP durch eine Headless-CMS-Konfiguration ersetzen, ist es wichtig, zunächst die täglichen Aufgaben Ihrer Redakteure, Marketing-Mitarbeiter und anderer Benutzer zu identifizieren.

Erfassen Sie die Unterschiede bei wichtigen Funktionalitäten. Neben den Content Management Anforderungen, wie z.B. Rich-Content-Modelle und Workflows, sollten dazu auch Funktionen für periphere Site-Managementaufgaben zählen, wie z.B.:

- die Verwaltung von Navigation, Umleitungen und ähnliche Aufgaben,
- das Hinzufügen von Tracking-Skripten für Marketing, Usability-Tracking usw.,
- die Benutzerverwaltung mit dem eigenen Benutzer-Backend,
- die Planung von Wochenend-Werbeaktionen,

- das Ändern von Landingpage-Strukturen,
- die Vorschau von Inhalten und eine Sammlung von Content-Objekten.

All dies kann entweder mit einem integrierten System oder mit einer Kombination aus Content-API und Anzeigeschicht erreicht werden. Allerdings benötigen Sie für letzteres möglicherweise Tools von Drittanbietern und zusätzliche Integrationen.

Betrachten Sie Produktfunktionen nicht als selbstverständlich, wenn Sie von einer integrierten Software-Suite zu einer Microservices-Architektur wechseln, die aus hochspezialisierten Einzelkomponenten aufgebaut ist.



# Komplexität und Robustheit der Implementierung

Eine Kombination aus den richtigen Fähigkeiten und Technologien ist gefragt

Aufgrund ihrer Reife lassen sich traditionelle CMS-Implementierungen großer Projekte einfacher abschätzen als eine Headless-Implementierung mit ähnlichem Umfang. Das liegt nicht daran, dass die Technologie besser oder schlechter ist, sondern daran, dass die Arbeitsweisen und die Erwartungen an die Fähigkeiten des Systems und die Aufgabenverteilung besser bekannt sind. Schätzungen sind daher realistischer, weil sie auf Erfahrungen, statt auf Vermutungen basieren.

Bei komplexen Headless-Implementierungen ist dies oft nicht so. Häufig ist es die erste Implementierung dieser Art für den Kunden, und der Integrationspartner verfügt u.U. nicht über ausreichend Erfahrung. Dies kann zu Pannen und unerwarteten Kosten während und nach dem Projekt führen, selbst wenn Sie die im zweiten Abschnitt dieses E-Books beschriebenen Funktionen mit der gebotenen Sorgfalt geprüft haben. Gut verwaltet, sind diese kein Problem.

Ein wesentliches Merkmal eines CMS-Projekts

ist, dass es sich in der Regel um ein Multivendor-Projekt handelt, bei dem mehrere Technologien zum Einsatz kommen. Während einige CMS-Anbieter Implementierungsprojekte selbst durchführen, ist es recht selten, dass ein CaaS-Anbieter (Content as a Service) Implementierungsservices anbietet.

Mit dem Geschäftsmodell eines API-Anbieters ist es allerdings nicht vereinbar. Wie bei jedem Technologieprojekt sollten Sie daher die Erfahrungen und Referenzen eines Implementierungspartners mithilfe der Content-API Ihrer Wahl prüfen – Sie wollen ja nicht sein erster Versuch sein.

Die Integrationsarchitektur ist ein weiterer Aspekt, der zu bedenken ist. CMS sind traditionell serverseitige Systeme, bei denen Verbindungen zu externen Systemen auf dem Server aufgebaut sind.

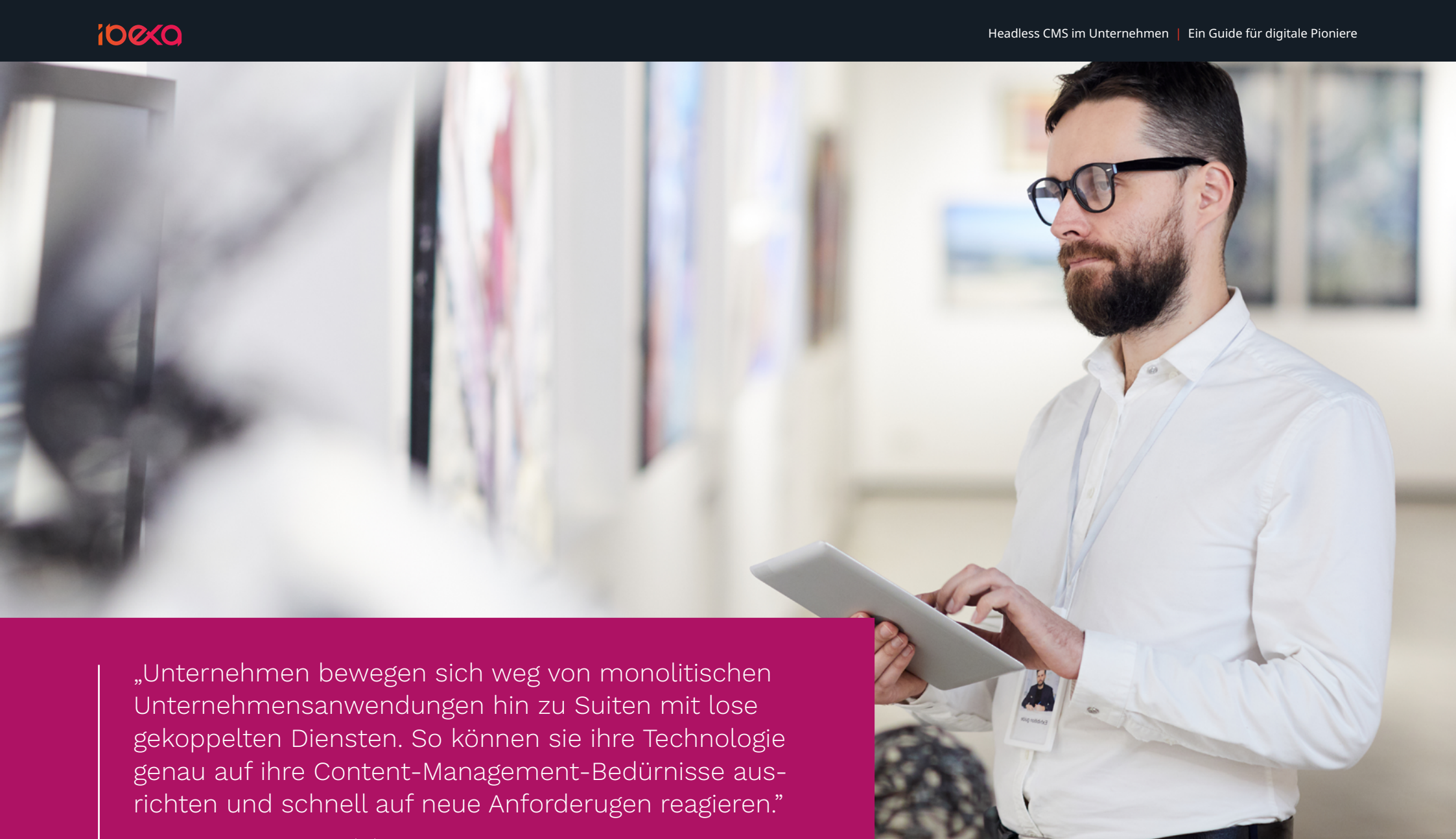
Der Trend geht aktuell dahin, die Integrationen auf den Client zu verlagern – den Browser. Das hängt nicht direkt damit zusammen, ob es um ein

Headless CMS geht oder nicht. Beides fällt jedoch oft zusammen, da es sich in beiden Fällen um neuartige Technologien handelt.

Den technischen Architekturansatz, bei dem der Browser zur Integrationsplattform wird, nennt man auch JAMStack.

Der Begriff leitet sich ab von JavaScript, API und Markup (JAM). Die Kernidee ist, Best-of-Breed-API-Anbieter für Authentifizierung, E-Commerce, Zahlungen usw. ausfindig zu machen und im Browser zu integrieren, um den Betrieb serverseitiger Systeme zu vermeiden.

Im besten Fall bietet eine JAMStack-Implementierung die Flexibilität, die beste Technologie zu niedrigen Kosten zu wählen. Im schlimmsten Fall ist es ein fragiles Durcheinander von Diensten und es kann zeitraubend sein, Probleme zu identifizieren. Die Realität findet sich wahrscheinlich irgendwo dazwischen.



„Unternehmen bewegen sich weg von monolithischen Unternehmensanwendungen hin zu Suiten mit lose gekoppelten Diensten. So können sie ihre Technologie genau auf ihre Content-Management-Bedürfnisse ausrichten und schnell auf neue Anforderungen reagieren.“

Demian Hess Enterprise Knowledge





# Projekt- und langfristige Wartungskosten

## Anschaffungs- und Betriebskosten im Auge behalten

Die Investitionen für eine Implementierung laufen auf die Kosten der Content-API und des API-Nutzer) hinaus. Die Gesamtkosten fallen also während der anfänglichen Entwicklungsphase und über mehrere Jahre der Entwicklung und Wartung an.

Bei der Content-API ist einfach. Sie müssen nur die auf dem Markt verfügbaren Optionen berücksichtigen und diejenige auswählen, die Ihnen die erforderlichen Kapazitäten und Funktionen zum günstigsten Preis bietet – ein reines Zahlenspiel.

### Bei der Auswahl einer Content-API sollten Sie mindestens die folgenden Punkte beachten:

- Kapazität (Anzahl der Contentobjekte, Repositories, Redakteure...)
- Verfügbare Funktionen vs. kundenspezifische Entwicklung
- Hosting-Kosten (für CMS-Anbieter)
- Abonnementkosten (für CaaS)

Hier gilt es, die Lösung mit der besten Balance für Ihre individuellen Bedürfnisse zu finden.

Die Preisgestaltung eines CaaS-Anbieters kann z.B. bei geringem Volumen sehr attraktiv sein, aber in der Größenordnung, die Ihr Unternehmen benötigt, in die Höhe schnellen. Die wegfallenden Lizenzgebühren eines Open-Source-Systems können wiederum durch höhere Hosting-Kosten aufgewogen werden. Achten Sie auch unbedingt auf mögliche zukünftige Kosten, falls Sie weiterwachsen wollen.

Ein PaaS wie die Ibexa Cloud könnte andererseits einen Kostenvorteil gegenüber IaaS-Anbietern bieten, da sich die Personalkosten für ein eigenes internes Dev-Ops-Team erübrigen.

Nach der Wahl des Content-API-Anbieters ist es an der Zeit, sich mit der Anzeigeschicht zu beschäftigen. Hier steht eine quasi unbegrenzte Anzahl von Optionen zur Verfügung. Von maßgeschneiderten Erstellungen über produktbasierte Implementierungen auf der

Framework-Grundlage wie Next.js oder Nuxt.js bis hin zu Static Site Generators wie Gatsby.js oder Hugo – praktisch alle diese Tools sind Open Source, so dass sich die Kosten nach dem Arbeitsaufwand richten. Achten Sie auch hier bei der Auswahl des Implementierungspartners auf die Erfahrung, da dies eine höhere Qualität und Entwicklungsgeschwindigkeit gewährleistet.

Bei den langfristigen Kosten kann es knifflig werden. Bei der Content-API können sie sich ändern – in der Regel sind sie aber recht stabil. Technisch gesehen könnte jedoch ein CaaS-Anbieter die Preise auf Basis der vereinbarten Abonnementbedingungen buchstäblich über Nacht anheben.

Dies ist jedoch ziemlich unwahrscheinlich, da es viele Alternativen gibt und der Wechsel zwischen den Anbietern relativ unproblematisch ist. Sie müssten Ihre Implementierung etwas überarbeiten, aber eine gut strukturierte Content-Anzeigeschicht („Head“) kann dies ohne allzu hohe Kosten bewältigen.





## Kontinuierlichen Wandel berücksichtigen Vorteile einer Microservices-Umgebung nutzen

Die Anzeigeschicht (API Consumer) wird bei einer Headless-CMS-Lösung zu einer strategischen Entscheidung. Bei traditionellen CMS sind die Templates und die Frontend-Implementierung hingegen oft austauschbar.

Bei Unternehmensimplementierungen, bei denen ein Web Content Management System (WCMS) nach der anfänglichen Implementierung noch fünf bis zehn Jahre im Einsatz sein kann, wird das Design mehrfach erneuert. Im Backoffice addieren sich die Funktionen immer weiter.

In einer integrierten End-to-End-Content-Management-Lösung sind die Content-API und die Anzeigeschicht naturgemäß eng miteinander gekoppelt – sie sind ein Teil eines Pakets.

Einer der Hauptvorteile der Headless-Architektur besteht darin, dass beide unabhängig voneinander sind. Es gibt jedoch eine natürliche Tendenz, dass sich die Investitionen für Features in der Anzeigeschicht akkumulieren, so dass das ursprüngliche Ziel, unabhängig zu sein, um eine der Komponenten auszutauschen, hinfällig wird.

Wenn Sie Ihre Implementierung eng an eine bestimmte Designschicht koppeln, müssen Sie am Ende mit steigenden Kosten rechnen. Im Laufe der Zeit häufen sich in der Anzeigeschicht die „technischen Schulden“ an. Das erfordert wiederum einen erhöhten Entwicklungsaufwand.

Ihre Fähigkeit, sich weiterzuentwickeln, wird beeinträchtigt, und es könnte sein, dass Sie erhebliches Rewriting betreiben müssen. Man kann

leicht der Illusion verfallen, ein entkoppeltes System zu haben, nur weil man sich anfangs für eine High-Level-Architektur entschieden hat. Agil zu bleiben erfordert Bemühung.

Software veraltet zwangsläufig früher oder später. Aber Sie können ein System so gestalten, dass Bestandteile austauschbar sind.

Um die Flexibilität einer Headless-Implementierung aufrechtzuerhalten, ist es entscheidend, bewusst Funktionalitäten zu entwickeln, die frei verfügbar sind.

Teilen Sie hierzu komplexe Funktionen in wiederverwendbare Dienste auf, die nicht an eine bestimmte Anzeigeschicht gebunden sind. Auf diese Weise können Sie die Vorteile des entkoppelten Ansatzes über Jahre der Entwicklung hinweg nutzen.

**ibexa DXP wurde als Content-Management-Lösung für die digitale Plattform an Bord von Viking Lines implementiert. Die Inhalte von Ibexa DXP werden an Bord der Schiffe von Viking Line für mobile Anwendungen, Browser-UI, Kabinen-TV und Digital Signage Bildschirme angezeigt.**

# Zusammenfassung

Entscheidend ist die Investition in eine DXP mit leistungsstarken Content-Funktionen und offenen APIs

Für die technische Implementierung einer Unternehmens-IT-Lösung gibt es heute mehr Optionen denn je. Für ein Unternehmen, das mit einem schwerfälligen System arbeitet, dessen technische Lebensdauer das Mindesthaltbarkeitsdatum bei weitem überschritten hat, kann der Headless-Ansatz ein frischer Wind sein, aber es ist nicht unbedingt immer die beste Lösung.

Die Fülle an Optionen kann die Wahl der richtigen erschweren, und es kann leicht passieren, dass es am Ende nicht ideal passt.

Wenn Sie z.B. reines Multichannel-Publishing betreiben, benötigen Sie kein Site-Management und dann ist ein Headless CMS genau das Richtige für Sie. Zahlen Sie nicht für Funktionen, die Sie nicht benötigen.

Gleichzeitig ist zu bedenken, dass die Verfügbarkeit einer neuen Option bestehende Tools nicht überflüssig macht. Auch wenn bestimmte Funktionen veraltet

erscheinen, können sie bei der Implementierung eines Content-Management-Projekts unerlässlich sein – ein notwendiges Übel.

Für diese Zwecke können Sie Funktionen aus integrierten Angeboten wie DXPs verwenden oder ein reines Headless CMS mit integrierten Tools von Drittanbietern einsetzen, um die Lücken zu schließen.

Bei großen Projekten, die weitere Systeme hinzufügen, addieren sich Partner und Systeme und können die ursprünglich empfundenen Vorteile wieder eliminieren. Statt Änderungen schnell und zu geringeren Kosten umsetzen zu können, müssten Sie die gleiche Zeit und die gleichen Ressourcen für die Koordination mehrerer Systemanbieter und die Zahlung fragmentierter Gebühren an die Dienstleister aufwenden.

Dies gilt insbesondere dann, wenn man die Gesamtbetriebskosten einer Lösung über ihre gesamte Lebensdauer betrachtet.

Denken Sie daran, dass keine Technologie den Erfolg garantiert. Das Team und seine Kooperation entscheiden über die Umsetzung Ihres Projekts. Das gilt für kleine und große Vorhaben, aber die Erfolgchancen sinken tendenziell, je mehr Menschen und Technologien beteiligt sind. Ein sehr erfolgreicher Ansatz für ein fünfköpfiges Team kann bei einer Teamgröße von dreißig Personen zu entgegengesetzten Ergebnissen führen.

Weitere Einblicke in die [Vorteile für Marketing und Redaktionsteams](#) gibt es auf unserer Webseite.

Wenn Sie mehr über Headless CMS-Lösungen erfahren möchten oder sich gerade mitten in einem Evaluierungsprojekt befinden, fordern Sie eine Demo mit unseren erfahrenen Content-Management-Experten an. Gerne stehen wir Ihnen auch [E-Mail](#) oder [Telefon](#) zur Verfügung, um Ihre Anforderungen für eine erfolgreiche digitale Transformation in Ihrem Unternehmen zu erörtern.



## Über Ibexa

Die Digital Experience Platform (DXP) von Ibexa ermöglicht es Unternehmen, wettbewerbsfähig zu bleiben, ihre Einnahmequellen zu optimieren, neue Produkte und Dienstleistungen auf den Markt zu bringen und digitale Strategien umzusetzen. Mit der DXP von Ibexa können B2B-Unternehmen ihre Verkaufsstrategien umgestalten und reibungslose Käuferlebnisse auf nachhaltige, kostengünstige und schnelle Weise aufbauen. Ibexa bietet dabei Content-Erstellung, Website-Aufbau, E-Commerce und Personalisierung zusammen mit beschleunigten Entwicklungsmöglichkeiten.

Des Weiteren geht es darum für Nutzer unvergessliche Erlebnisse channel-übergreifend zu schaffen, indem Marke, Inhalt, Produktinformationen, Handel und ihre bestehenden Geschäftsprozesse vereinheitlicht werden. Mit Hilfe unserer zertifizierten, vertrauenswürdigen Partner - Digitalagenturen, Systemintegratoren und Beratungsfirmen - werden Unternehmen in ihrer digitalen Transformation voranschreiten und sich schnell an die sich ändernden Marktbedürfnisse anpassen, während sie das Kundenerlebnis stets in den Mittelpunkt ihres unternehmerischen Handelns stellen.

## Mehr erfahren?

## Nehmen Sie Kontakt mit uns auf:

### **Ibexa GmbH**

Kaiser-Wilhelm-Ring 30-32

50672 Köln

+49 (0) 221 3674 860

[www.ibexa.co/de](http://www.ibexa.co/de)

[info.de@ibexa.co](mailto:info.de@ibexa.co)



## Über elbformat

elbformat wurde 2009 gegründet und startete mit einem Team von 4 Mitarbeitern. Heute befinden sie sich im 10. Firmenjahr mit 35 Mitarbeitern und wachsen stetig weiter. Sowohl als Unternehmen als auch jeder Mitarbeiter für sich - Stehenbleiben gibt es bei ihnen nicht. Dafür aber regelmäßiges Reflektieren, Visualisieren und einfach Machen.

Die Implementierung von Enterprise Content Management Lösungen ist ihre Herkunft. Auf dieser Basis erschaffen sie Websites, Landingpage-Baukästen, Intranets und Digitale Styleguides. Mit ihrer jahrelangen Erfahrung unterstützen sie bei Betrieb und Hosting.

## Jetzt elbformat kontaktieren:

### **elbformat GmbH**

Großer Burstah 50-52

20457 Hamburg

+ 49 (40) 209 31 04 0

[www.elbformat.de](http://www.elbformat.de)

[info@elbformat.de](mailto:info@elbformat.de)