



# Kundenwebsites mit Jamstack erstellen: Schnell. Effizient. Sicher.

Wir zeigen Ihnen, welche Probleme konventionelle CMS mit sich bringen, warum statische Seiten mithilfe des Jamstacks zur echten Alternative werden – und wie Sie damit schlankere Workflows und bessere Kundenergebnisse gewährleisten.

# Inhalt

<b>1. Erkennen Sie sich wieder?</b>	<b>3</b>
Code-Recycling, Livetests, Feedbackschleifen – der klassische Dreiklang der Webentwicklung?	
<b>2. Ein Blick zurück: Dynamische Websites = WordPress, Joomla &amp; Co.</b>	<b>3</b>
Websites werden Interaktiver und dynamischer – zwangsläufig nur mit LAMP-Stack-basierten CMS?	
<b>3. Konventionelle CMS: Ein Modell für die Zukunft?</b>	<b>4</b>
Komplexe Datenbankstrukturen und ein hoher Ressourcenbedarf. Die Basis für moderne Websites?	
<b>4. Statische Websites versprechen Caching, hohe Fehlertoleranz und Sicherheit</b>	<b>5</b>
Geringe Ladezeiten und eine hohe Verfügbarkeit entscheiden über den Erfolg einer Website.	
<b>5. Statisch meets Jamstack: Das Erfolgsrezept für konkurrenzlos gute Websites</b>	<b>6</b>
Eine Webarchitektur auf Basis von Entkopplung und Prerendering geht Probleme klassischer statischer Websites gezielt an – und macht sie zu einer echten Alternative.	
<b>6. Auf Entwickler ausgerichtet: Jamstack definiert Workflows völlig neu</b>	<b>9</b>
Die enge Integration von GitHub & Co. ermöglicht einen reibungslosen Entwicklungsprozess mit einer einfachen Versionisierung.	
<b>7. Der Jamstack: Vorteile und Entwicklungspotenziale für Agenturen</b>	<b>10</b>
Bis zu 10x schneller, beliebig skalierbar und ein geringer Wartungsaufwand – der Jamstack als Benchmark für Webprojekte der Zukunft.	
<b>8. Vielfältig einsetzbar: Die Toolchain von Jamstack</b>	<b>12</b>
Eine wachsende Anzahl an Programmierwerkzeugen schafft viele Optionen, um attraktive Websites zu gestalten.	
<b>9. Deploy Now: Performantes Hosting für Ihre Jamstack Webseite</b>	<b>13</b>
Bringen Sie Ihre Jamstack Website ganz einfach von Ihrem GitHub Repository auf die IONOS-Infrastruktur. Ihr Deployment für verschiedene Branches, jederzeit georedundant.	
<b>10. Kundenprojekte mit Deploy Now: In wenigen Schritten ans Ziel</b>	<b>13</b>
Framework auswählen, Git Setup aufbauen, ... – so einfach entwickeln Sie Ihre Webprojekte mit Deploy Now.	

## 1 Erkennen Sie sich wieder?

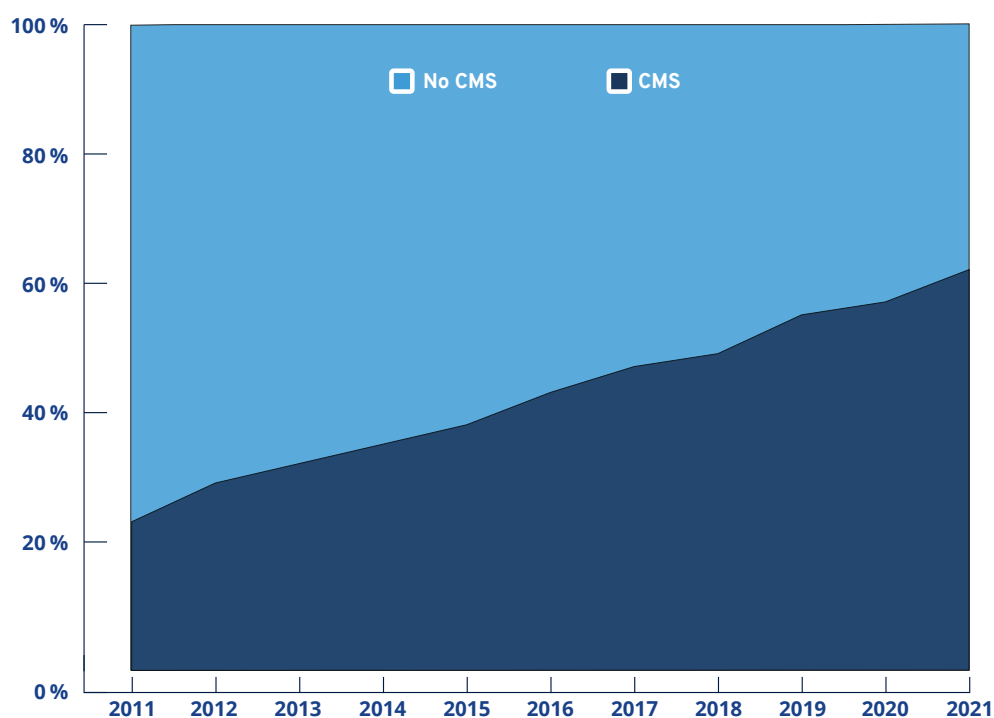
Als Web Professional realisieren Sie unterschiedlich komplexe Webprojekte für Ihre Kunden. Typischerweise greifen Sie dabei auf bestehenden Code zurück, passen diesen an das Projekt an und nutzen aufgrund der Einfachheit ein Content-Management-System (CMS) wie WordPress, Joomla oder Drupal. Der Entwicklungsprozess ist geprägt von einer Reihe von Livetests, gefolgt von kundenseitigem Feedback.

**Trifft diese Beschreibung auf Sie zu?** Dann können Sie mit dem Umstieg auf Jamstack schon jetzt die Weichen für die Zukunft stellen.

## 2 Ein Blick zurück: Dynamische Websites = WordPress, Joomla & Co.

Als das World Wide Web vor 30 Jahren mit der [ersten Website](#) gestalt annahm, dominierten statische Seiten viele Jahre unser Bild vom Internet. Ein bisschen HTML, garniert mit CSS, an jeden User so ausgeliefert, wie die Seite erstellt wurde – fertig war das Webprojekt.

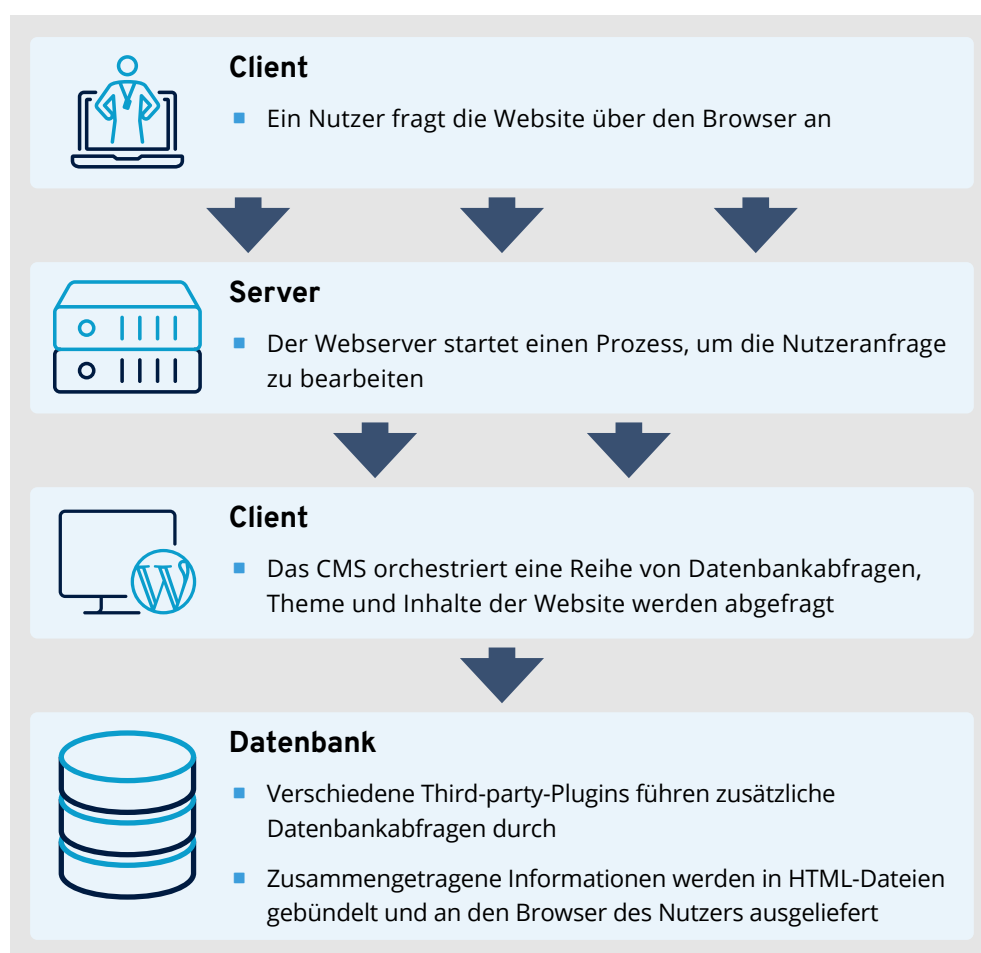
Um die Jahrtausendwende entstand mit Web 2.0 ein anhaltender Trend, der immer mehr interaktive und soziale Elemente hervorbrachte. Das Web wurde dynamisch, Content-Management-Systeme wie WordPress versprachen dem User erstmalig die einfache Erstellung und Verwaltung von Webinhalten – ganz ohne Programmierkenntnisse.



**Das Resultat:** Die Anzahl an Websites stieg exponentiell, auch kleine Unternehmen, Vereine oder Privatpersonen waren in der Lage, online zu gehen. Im Wettbewerb um Besucher rückte eine auf zentrale Keywords ausgerichtete Suchmaschinenoptimierung in den Fokus.

### 3 Konventionelle CMS: Ein Modell für die Zukunft?

Während statische Webseiten aus einer simplen Datenstruktur bestanden, die stets bereit für die Auslieferung auf dem Webserver verfügbar sind, verbirgt sich hinter konventionellen CMS eine komplexe Architektur mit zeitraubenden Antwortprozessen:



Dieser Prozess unterlag in den letzten 20 Jahren praktisch keiner technischen Veränderung. Sind LAMP-Stack-basierte Content-Management-Systeme damit die Wahl die Zukunft? Gewisse Zweifel sind angebracht, denn die Erfolgskriterien moderner Websites (geringe Ladezeit, Skalierbarkeit, etc.) sind klar auf Performance ausgerichtet.

### **Ladezeit**

Die Summe an notwendigen Datenbankabfragen lässt die Ladezeit einer CMS-basierten Website steigen. Um diesen Nachteil auszubessern, werden häufig Caching-Mechanismen entwickelt, was die Komplexität der Architektur zusätzlich erhöht. Langsame Ladezeiten werden heute von Nutzern kaum mehr toleriert. Schlechte Ladezeiten wirken sich zudem auf das Suchmaschinen-Ranking aus.

### **Ressourcenverbrauch**

Das Setup eines CMS bedarf in aller Regel zwei Ressourcen: einem WebServer mit Dateisystem und einer Datenbank. Da Anfragen an ein CMS häufige DB-Abfragen erzwingen, wird häufig die DB zum kritischen Flaschenhals. Der Ressourcenhunger schlägt sich bei erfolgreichen Websites schnell in höheren Infrastrukturkosten nieder.

### **Skalierbarkeit**

In Zeiten von Influencern, viralen Kampagnen und Newslettern können Trafficzahlen sprunghaft in die Höhe steigen. Durch solche Traffic-Spitzen sind Content-Management-Systeme schnell überlastet. In der Folge werden Anfragen in Warteschlangen eingereiht oder sofort abgebrochen.

### **Wartung und Sicherheit**

Die Versionen eines CMS und dessen Plugins müssen regelmäßig aktualisiert werden, um bekannte Sicherheitslücken zu schließen. Zudem bietet die Architektur eines CMS durch die serverseitigen Prozesse und die Datenbank vielfältige Angriffsflächen für Hacker.

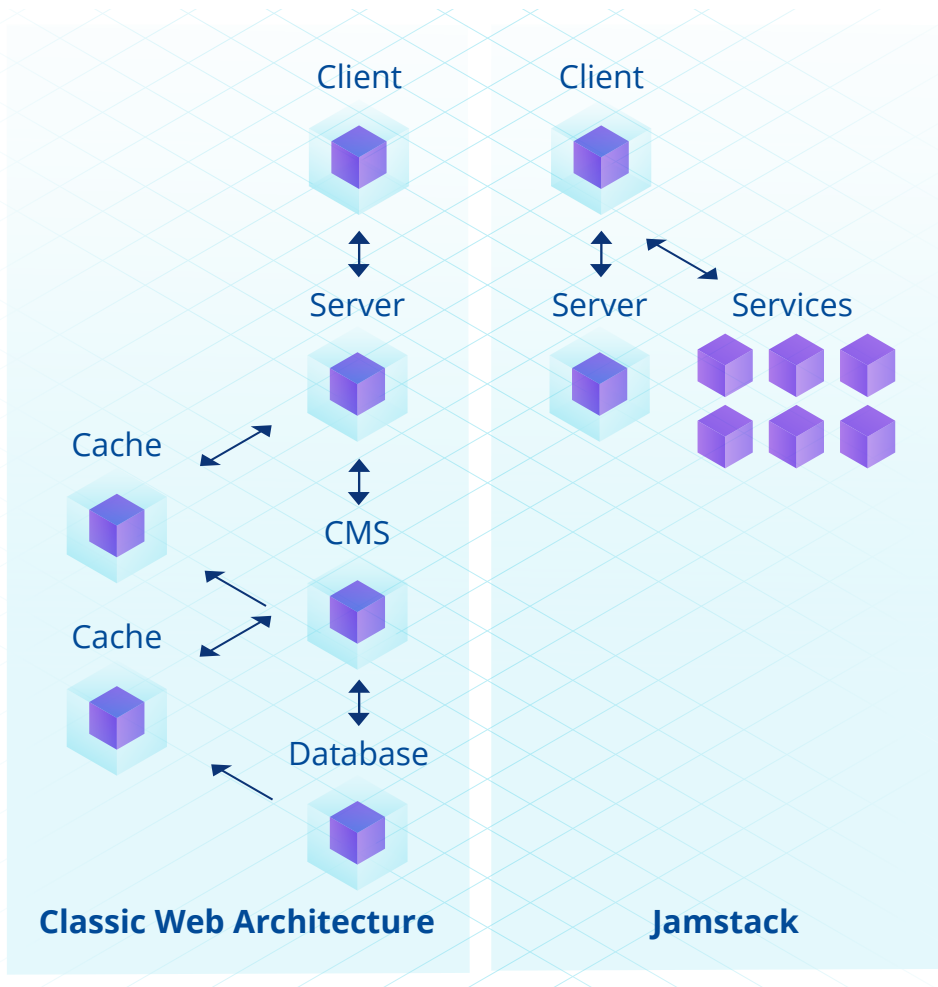
## **4 Statische Websites versprechen Caching, hohe Fehlertoleranz und Sicherheit**

Statische Websites sind zu 100% cachebar. Attraktiv sind für Sie besonders Content Delivery Networks, sprich geographisch verteilte Caches, die Ladezeiten auf ein Minimum reduzieren. Zwingend notwendig sind Caches für statische Webseiten aber nicht. Da der Webserver die existierenden HTML- und CSS-Dateien lediglich ausliefern muss, sind Ladezeiten kein reales Problem. Da statische Webseiten komplett ohne Skripte oder serverseitige Logiken funktionieren, ist ihre Fehlertoleranz hoch. Die Angriffsfläche auf statische Websites ist gering. Dennoch bleiben die Nachteile der statischen Seite, die in den letzten Jahren dem CMS zum Aufstieg verholfen hat. Statische Seiten bieten per se keine Schnittstelle, über die Editoren leicht Inhalte erstellen können, ohne in den Code zu gehen. Sie erhalten sehr oft duplizierten Code, der sich unverändert in einer Vielzahl von HTML-Dateien wiederfindet. Und es gibt keine Möglichkeit, externe, veränderliche Daten in Webseiten einzubinden.

## 5 Statisch meets Jamstack: Das Erfolgsrezept für konkurrenzlos gute Websites

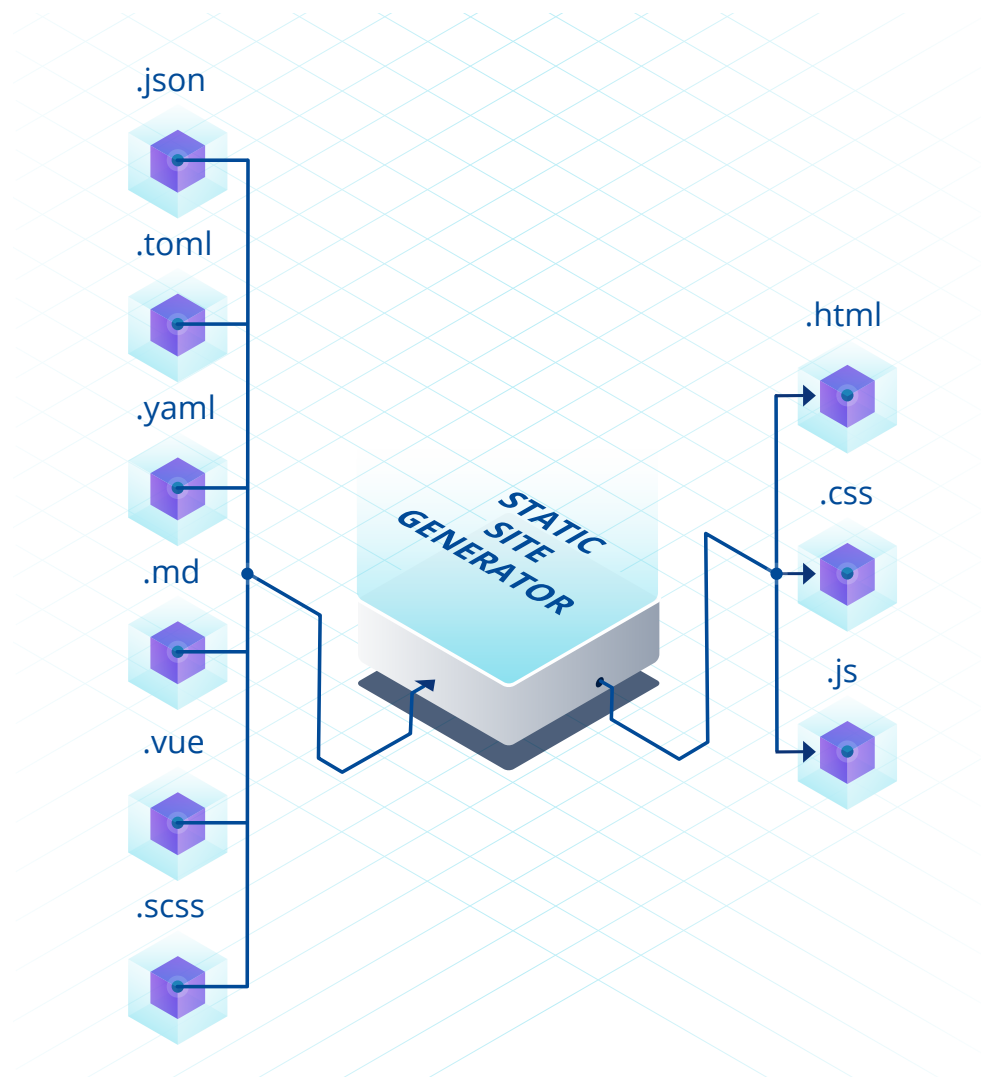
Mit dem Jamstack wird die statische Website wieder zur echten Alternative. Jamstack steht für clientseitiges JavaScript, wiederverwendbare APIs und statisches Markup. Der Jamstack repräsentiert dabei vielmehr eine neue Webarchitektur, als einen konkreten Technologiestack. Die Kernprinzipien des Jamstacks sind Entkopplung und Pre-rendering.



**Entkopplung:**

Während die klassische Webarchitektur auf einen vielschichtigen Request-Response Prozess setzt, versucht der Jamstack den Antwortprozess möglichst schlank zu halten. Der Webserver liefert auf Anfrage direkt HTML- und CSS-Dateien an den Aufrufenden. Der Jamstack verzichtet dabei vollständig auf Datenbankabfragen. Der Webserver beschäftigt sich ausschließlich mit der Auslieferung statischer Inhalte, ohne Antworten an den Client zur Laufzeit zu berechnen. Dynamische Funktionalitäten werden nach der Auslieferung des Frontends über den Browser des Users aufgerufen. Hierfür wird clientseitiges JavaScript verwendet, das über APIs eine Vielzahl selbst entwickelter oder Third-party-Services anbindet.

## Prerendering:



Der sogenannte Static Site Generator stellt die zentrale Komponente des Jamstacks dar. Ein Static Site Generator ist ein Skript, das bei jeder Änderung einer Input Ressource eine Menge an statischer HTML Dateien neu generiert. Inzwischen steht für jede populäre Skriptsprache mindestens ein Static Site Generator zur Verfügung. Input-Ressourcen können beliebige Daten, Inhalte und Templates sein. Die generierten Dateien werden also nach jeder Änderung des Contents aktualisiert auf dem Webserver verfügbar gemacht. Dieser Aktualisierungsvorgang dauert vergleichsweise länger als das Abspeichern einer Contentänderung mit einem CMS. Dafür reduziert sich jedoch die Antwortzeit erheblich, da die statischen Dateien nicht erst auf jede Anfrage hin generiert werden müssen.





## 6 Auf Entwickler ausgerichtet: Jamstack definiert Workflows völlig neu

Ein großer Vorteil für Agenturen ist, dass sich der Jamstack als Lösung sehr stark an den Bedürfnissen von Entwicklern orientiert. Somit ist die enge Integration von Codeversionierungsplattformen wie GitHub, Bitbucket oder GitLab bei fast allen Jamstack-Hosting Providern Standard. Somit stellt der Livegang von Webprojekten keinen Bruch im Entwicklungsablauf mehr da.

Stattdessen können Sie schon während Ihrer Test- und Feedbackphase Webseiten direkt von einem Git Repository auf eine Liveumgebung deployen. Bei jeder Änderung von Code oder Content stößt der Static Site Generator einen Aktualisierungsprozess an und Sie können die Änderungen wenige Sekunden später live prüfen. Die meisten Anbieter ermöglichen hier das Deployment mehrerer Codebranches unter separierten Vorschau-URLs, falls Änderungen nicht direkt unter der Haupt-Domain sichtbar sein sollen.

**Das Ergebnis:** So können mehrere Entwickler leichter kollaborieren. Gleichzeitig ersetzt die Versionierungsfunktion von GitHub und Co. die Notwendigkeit von Webspacebackups.

“  
*Ich möchte mich eigentlich auf die Entwicklung und Gestaltung konzentrieren. Am besten verknüpfe ich einfach ein Repository und mein Hoster kümmert sich um den Rest.*  
”



**Manuel Kling,**  
Geschäftsführer  
Fachwerk Media GbR

## 7 Der Jamstack: Vorteile und Entwicklungspotenziale für Agenturen

### Vorteile

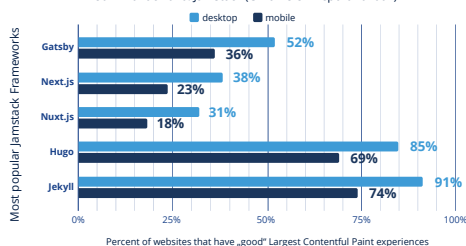


#### Bis zu 10x schnellere Websites

- Hohe Wahrscheinlichkeit für schnellen SEO-Erfolg
- Höherer Umsatz durch niedrigere Bounce Rates

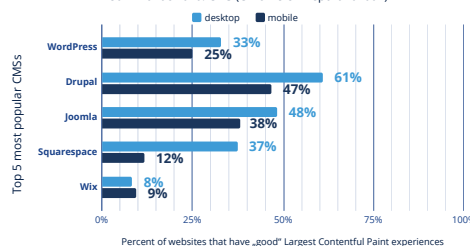
#### Real-user Largest Contentful Paint experiences

Web Almanac 2020: Jamstack (Chrome UX Report 202007)



#### Real-user Largest Contentful Paint experiences

Web Almanac 2020: CMS (Chrome UX Report 202007)



Der Largest Contentful Paint misst, wie früh der Hauptinhalt einer Website geladen ist. Vor allem die Static Site Generatoren Hugo und Jekyll können hier überzeugen.

Quellen: <https://almanac.httparchive.org/en/2020/cms>, <https://almanac.httparchive.org/en/2020/jamstack>

#### Geringere Sicherheitslücken, weniger Wartungsaufwand

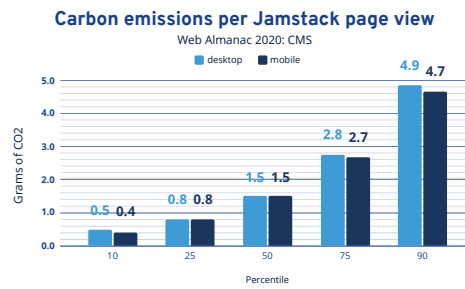
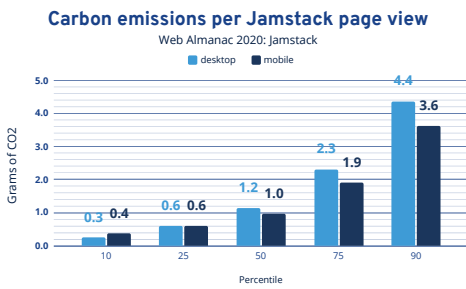


- Mit jedem zusätzlichen Plugin erhöht sich der Angriffsvektor einer klassischen CMS Website. Mit dem Jamstack reduziert sich dieser auf ein Minimum.
- **56% der Malwareinfektionen sind auf nicht durchgeführte Updates zurückzuführen** (<https://sucuri.net/reports/2019-hacked-website-report/>), welche Sie sich mit dem Jamstack weitestgehend sparen.

#### Geringerer Ressourcenverbrauch: Geringere Kosten, mehr Skalierbarkeit und besser für die Umwelt



- Da der Webserver kaum Rechenzeit verbraucht, fallen kaum Serverkosten an. Vor allem in Kombination mit einem gut aktualisierbaren Cache oder CDN können Sie Jamstackwebsites hervorragend skalieren.



Der Jamstack senkt den CO2 Fußabdruck von Websites erheblich

Quellen: <https://almanac.httparchive.org/en/2020/cms>, <https://almanac.httparchive.org/en/2020/jamstack>

“  
Ein schneller, sicherer  
und skalierbarer Online-  
Shop ist heutzutage  
enorm wichtig. Mit  
Jamstack haben wir die  
perfekte Grundlage,  
um diese Ziele zu  
übertreffen!



### Attraktiv für Talente

- Entkopplung und Git Integration erlauben moderne Entwicklungsprozesse mit agilen, autonomen Teams. Entwickler können auf modernste Frameworks zurückgreifen.



Enrico Franke,  
Geschäftsführer  
Fruchtec GbR

## Entwicklungspotenzial



### Integration dynamischer Funktionalitäten

- Das Einbinden eines Kontaktformulars oder gar Shops als Microservice bietet viele architekturelle Vorteile, ist aber mit deutlich mehr Integrationsaufwand verbunden als bei einem konventionellen CMS.



### Komfortables Erstellen von Inhalten

- Sogenannte Headless CMS sind sehr gut geeignet, um Jamstack Websites über ein übersichtliches Interface mit Inhalten zu befüllen. Allerdings können diese nicht mit der What-you-see-is-what-you-get-Erfahrung von WordPress und co aufwarten.



### Optische Anpassungen nur via Code

- Um die Optik einer Jamstack Website anzupassen, muss dessen Code angefasst werden. Dies führt gegebenenfalls zu einem erhöhten Arbeitsaufwand für Softwareentwickler.

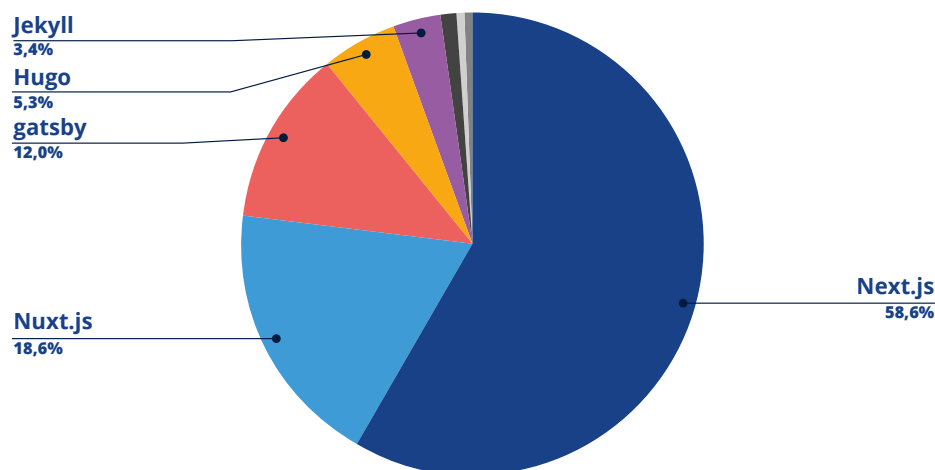
## 8 Vielfältig einsetzbar: Die Toolchain von Jamstack

Die Komplexität einer Jamstack Toolchain ist nicht zu unterschätzen. Dennoch bietet der Jamstack schon jetzt eine Vielzahl an Optionen, um Ihren Bedürfnissen gerecht zu werden.

Tool	Wofür?	Anbieter
JavaScript Frameworks	Moderne, komponentenbasierte Frameworks für das Frontend Ihrer Jamstackseite.	<ul style="list-style-type: none"> <li>Vue</li> <li>React</li> <li>Angular</li> </ul>
Static Site Generators	Generierung von statischen Dateien auf Basis verschiedener Inputs.	<ul style="list-style-type: none"> <li>Jekyll</li> <li>Hugo</li> <li>Nuxt</li> <li>Gatsby</li> </ul>
Headless CMS	Einfache Contentverwaltung über ein grafisches Interface.	<ul style="list-style-type: none"> <li>Contentful</li> <li>Forestry</li> <li>Sanity</li> </ul>
Git	Speicherung von Code und Assets inklusive Kollaboration und Versionierung.	<ul style="list-style-type: none"> <li>Github</li> <li>GitLab</li> <li>Bitbucket</li> </ul>
Ecommerce APIs	Verwandlung von Jamstack Websites in Onlineshops durch die Anbindung von Microservices.	<ul style="list-style-type: none"> <li>Stripe</li> <li>Snipcart</li> </ul>
Formulare	Einfache Integration von Kontaktformularen ohne Datenbanken.	<ul style="list-style-type: none"> <li>Typeform</li> <li>Formspree</li> </ul>

### Jamstack adoption share

Web Almanac 2020: Jamstack (Mobile)



**Beliebtste Site Generators**

Quelle: <https://almanac.httparchive.org/en/2020/jamstack>

## 9 Deploy Now: Performantes Hosting für Ihre Jamstack Webseite

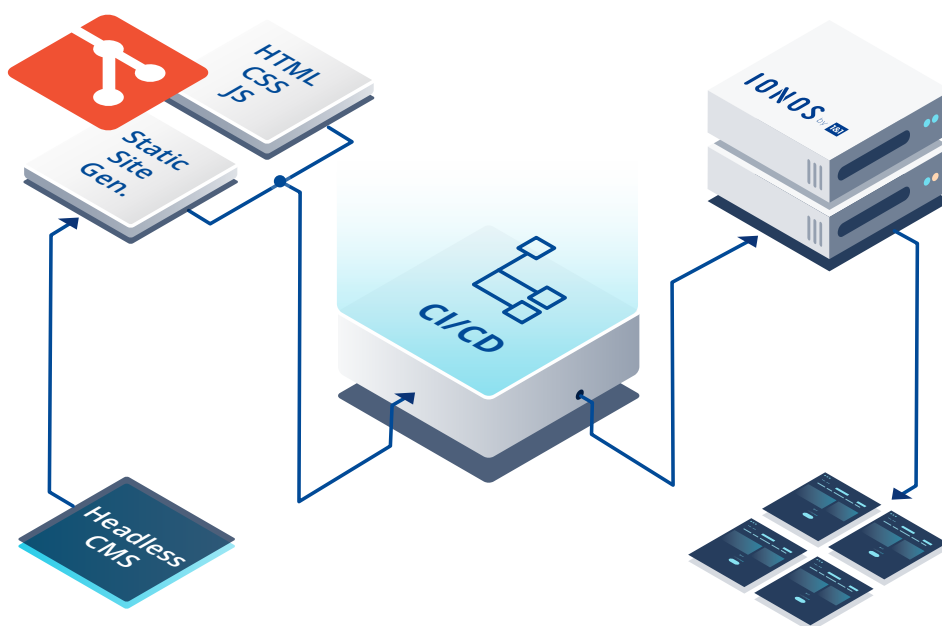
Mit Deploy Now können Sie Ihre Jamstack Website bequem von Ihrem GitHub Repository auf die IONOS Infrastruktur deployen. Ihre ersten drei Projekte sind dabei dauerhaft kostenfrei. In jedem Projekt können Sie sowohl produktive Branches deployen und mit Ihrer Domain verknüpfen, als auch Feature Branches anlegen und als Stagingumgebung nutzen. All Ihre Deployments sind dabei georedundant und sowohl DDoS- als auch SSL-geschützt.



# Deploy Now

Jetzt kostenfrei  
ausprobieren

## 10 Kundenprojekte mit Deploy Now: In wenigen Schritten ans Ziel



### Schritt 1: Das passende Framework auswählen

Suchen Sie sich den Static Site Generator aus, der zu Ihnen passt.

[Hier](#) finden Sie Generators für alle gängigen Skriptsprachen wie JavaScript, Go, Ruby oder Python. Viele JavaScript Generators orientieren sich dabei zusätzlich an Angular, React und Vue.

## Schritt 2: Git Setup aufbauen

Idealerweise sind alle an der Entwicklung beteiligte Mitarbeiter in der Lage, lokal am Websitecode zu arbeiten und diesen regelmäßig mit einem gemeinsam verwalteten Git Repository zu synchronisieren. Idealerweise öffnen Entwickler eigene Branches, um konfliktfrei an einzelnen Features zu arbeiten. [Hier](#) erfahren Sie, wie Ihren GitHub Workflow schnell aufsetzen können.

## Schritt 3: Startpunkt auswählen

Webprojekte beginnen selten von Null. Das Klonen von existierenden Git Repositories können einen schnellen Einstieg ins neue Webprojekt bieten. Sie können beispielsweise eigene Starter-Repositories entwickeln, Vorlagen aus dem Web suchen oder mit einem der [Deploy Now Starter](#) aus unserer Dokumentation beginnen.

## Schritt 4: Projekt anlegen

Sobald Sie entschieden haben, mit welchem Repository Sie starten wollen, können Sie es komfortabel mit Deploy Now verbinden. Von nun an werden Änderungen an Ihrem Code direkt vom Static Site Generator registriert und in Form von HTML und CSS Dateien auf den Webspaces übertragen.

## Schritt 5: Weiterentwickeln im Staging

Idealerweise überprüfen Sie Änderungen an Feature Branches in Staging Deployments, bevor Sie diese auf die Live-Version Ihrer Website übertragen. So werden Fehler und Konflikte von vornherein vermieden. In Deploy Now erhalten Sie zu jedem Staging Deployment eine Vorschau-URL die Sie bequem an Kunden und Kollegen weiterleiten können, um Feedback einzusammeln.

## Schritt 6: Mergen auf die Production Branch

Sind Sie mit Ihren Codeänderungen zufrieden, können Sie diese bequem auf die Hauptversion Ihrer Website übertragen. Dies können Sie entweder tun, indem Sie Änderungen von Featurebranches in Git auf Ihren Hauptbranch übertragen oder indem Sie Ihren Featurebranch in Deploy Now als den neuen Hauptbranch definieren.

## Schritt 7: Domain verbinden und live gehen

Nun können Sie Ihre Vorschau-URL bequem durch eine eigene Domain ersetzen. Diese wird von Deploy Now automatisch mit einem SSL Zertifikat ausgestattet.

# Impressum

IONOS SE  
Elgendorfer Str. 57  
56410 Montabaur, Germany

## IONOS Kontakt

Telefon +49 (0) 721 170 5522  
E-Mail [info@ionos.de](mailto:info@ionos.de)  
Website <https://www.ionos.de>

## Vorstand

Hüseyin Dogan, Dr. Martin Endreß, Claudia Frese, Hans-Henning Kettler,  
Arthur Mai, Britta Schmidt, Achim Weiß

## Aufsichtsratsvorsitzender

Markus Kadelke

## Handelsregister

IONOS SE: Amtsgericht Montabaur / HRB 24498

## Umsatzsteuer-Identnummer

IONOS SE: DE815563912

# Copyright

Die Inhalte des Guides wurden mit größter Sorgfalt erstellt. Für Richtigkeit, Vollständigkeit und Aktualität keine Gewähr.

© IONOS SE, 2021

Alle Rechte vorbehalten – einschließlich der, welche die Vervielfältigung, Bearbeitung, Verbreitung und jede Art der Verwertung der Inhalte dieses Dokumentes oder Teile davon außerhalb der Grenzen des Urheberrechtes betreffen. Handlungen in diesem Sinne bedürfen der schriftlichen Zustimmung durch IONOS SE. IONOS SE behält sich das Recht vor, Aktualisierungen und Änderungen der Inhalte vorzunehmen.