

HEADLESS CMS

Überblick & Umsetzung

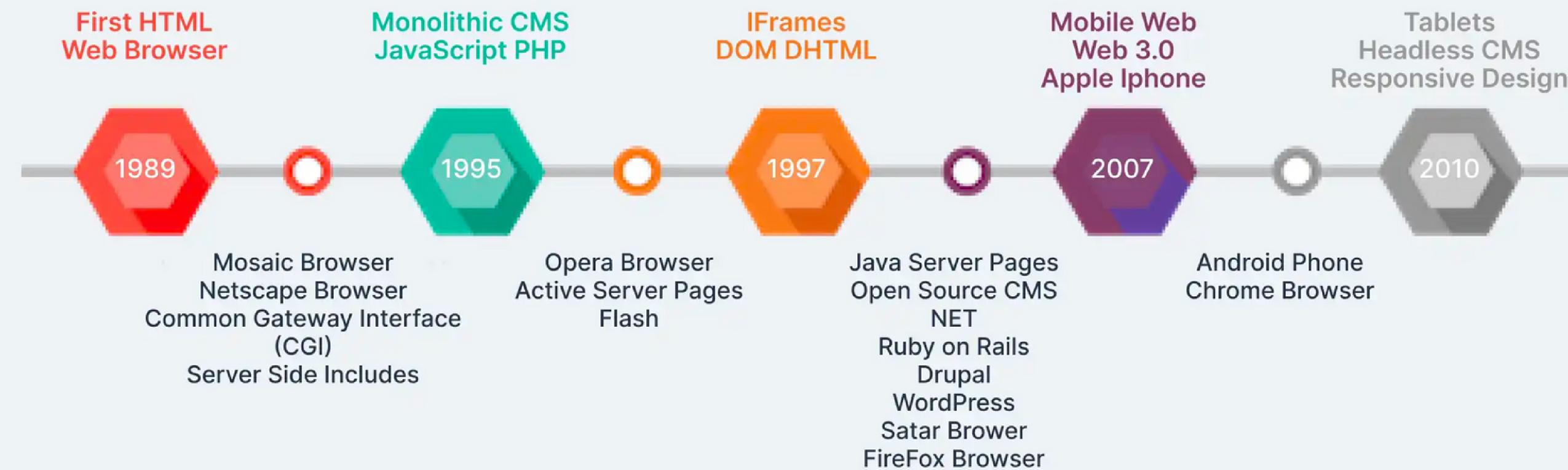
AGENDA

- EINFÜHRUNG
- VERGLEICH
- TRENDS
- VOR- & NACHTEILE
- FAZIT
- HANDS-ON

HISTORY

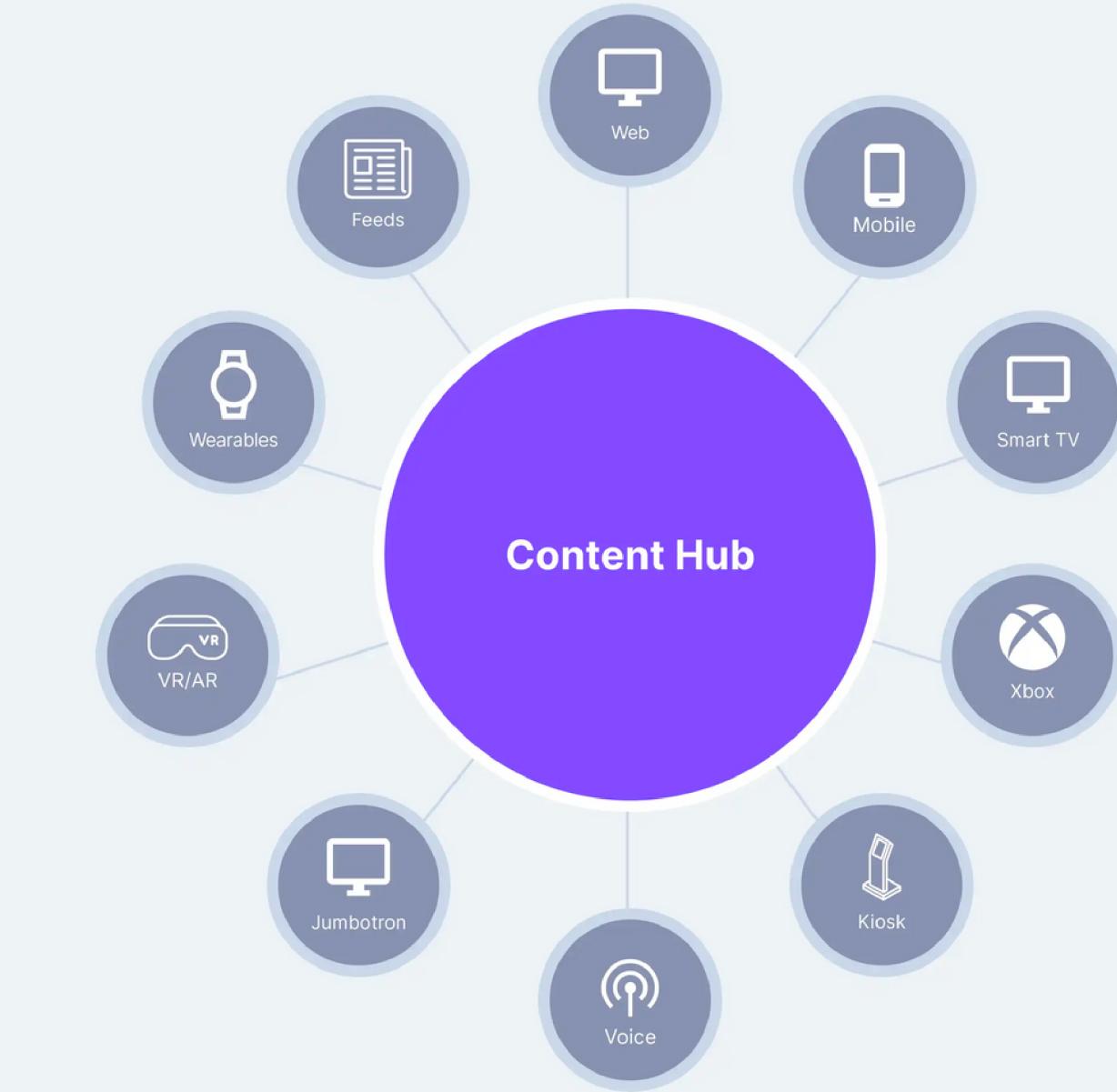
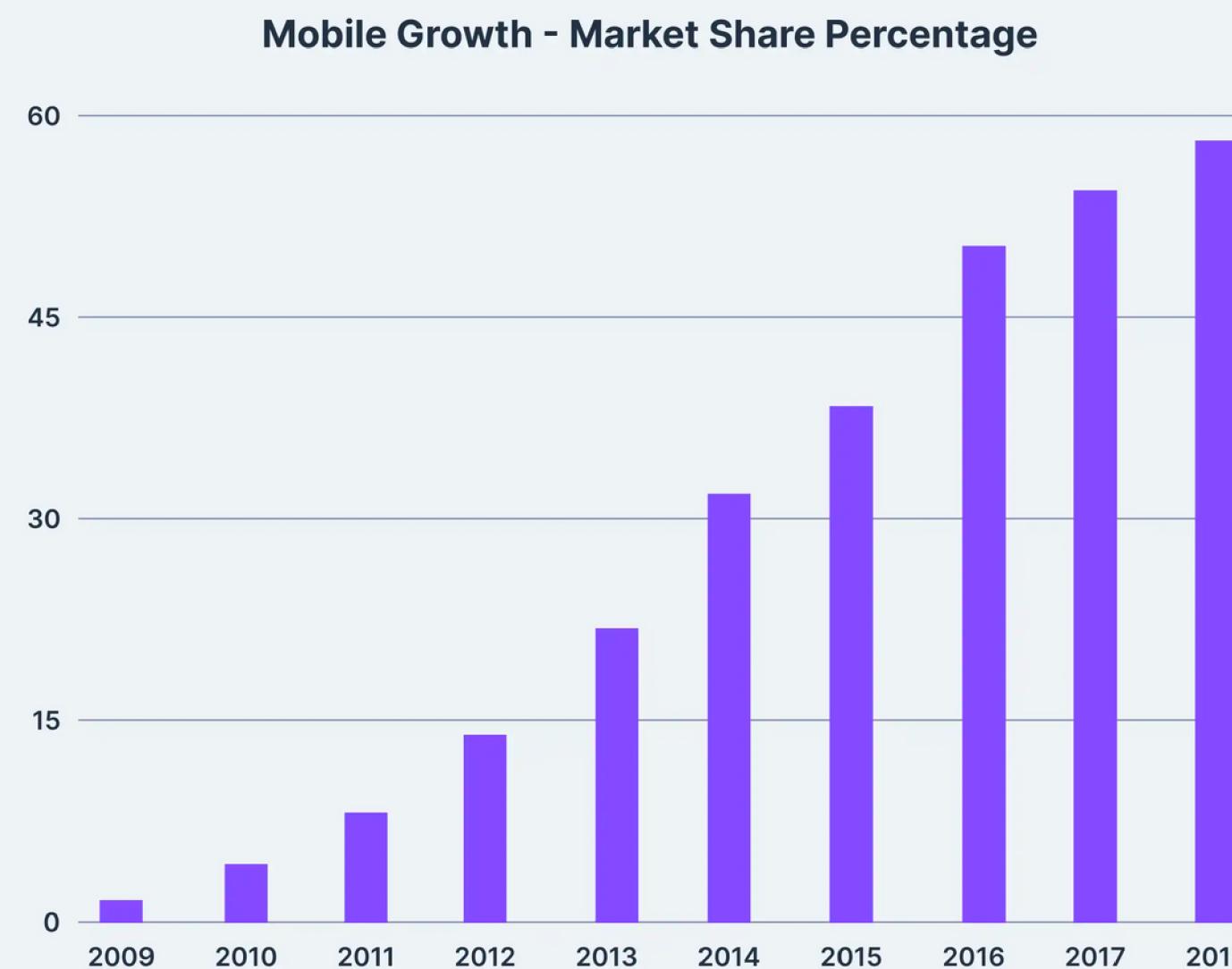
Entstehung des CMS im Web

Milestones for web content management



HISTORY

Vergleich traditionale CMS



VERGLEICH

Die beliebtesten Monolith CMS



Pros

- Easy to use
- Extensive plugin library
- Large community support

Cons

- Limited scalability for complex websites
- Security vulnerabilities



Pros

- Highly scalable
- Robust security features
- Suitable for complex websites

Cons

- Steeper learning curve
- Requires technical expertise



Pros

- Flexible and customizable
- Suitable for e-commerce websites
- Multilingual support

Cons

- Steeper learning curve
- Fewer plugins compared to WordPress

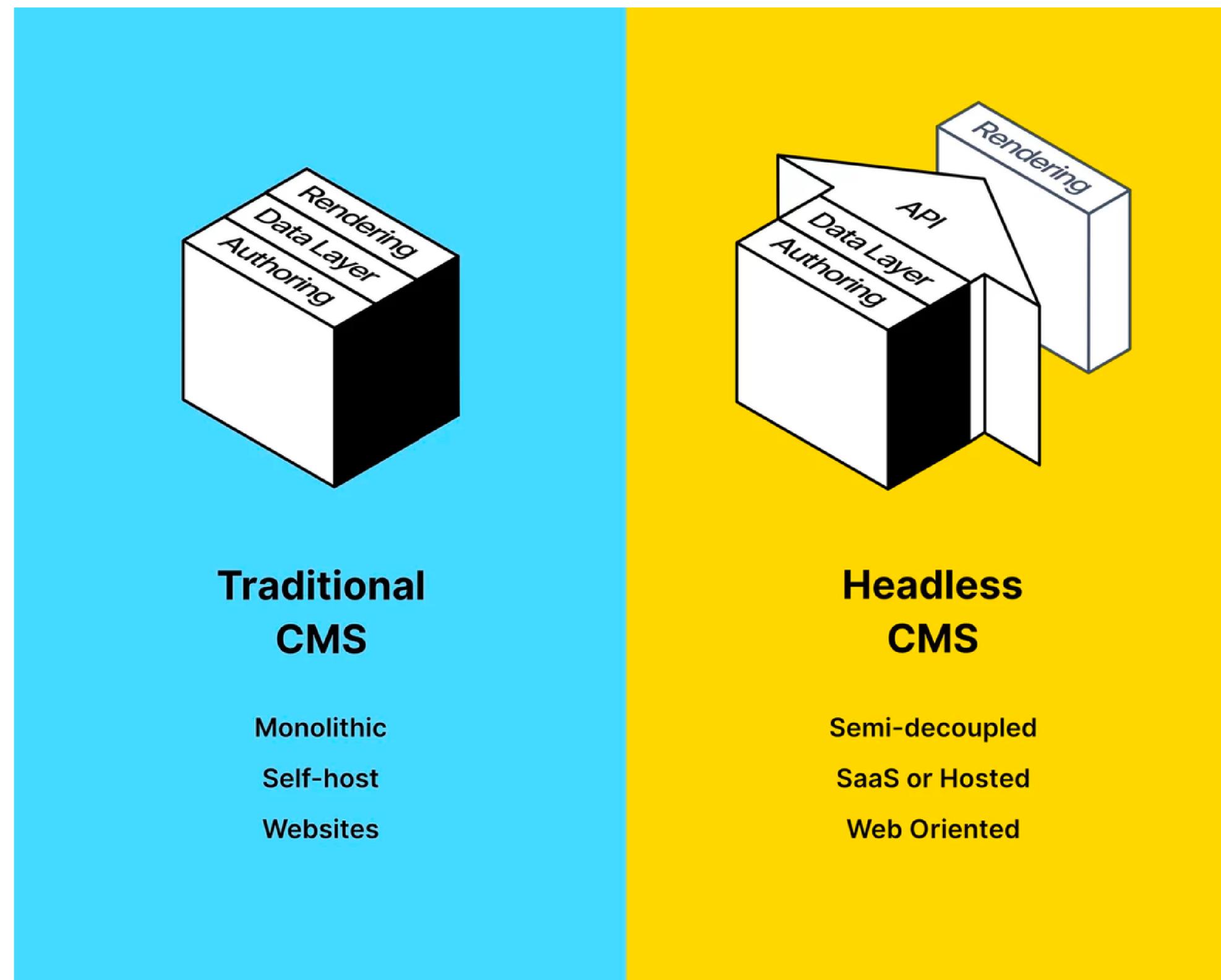
WAS?

Ein Headless CMS ist ...

ein **Backend-Only Web Content-Management-System (CMS)**. Ein Es stellt **Inhalte über eine API** für die Anzeige auf jedem Gerät bereit, **ohne ein integriertes Frontend** oder einen Presentation Layer.

VERGLEICH

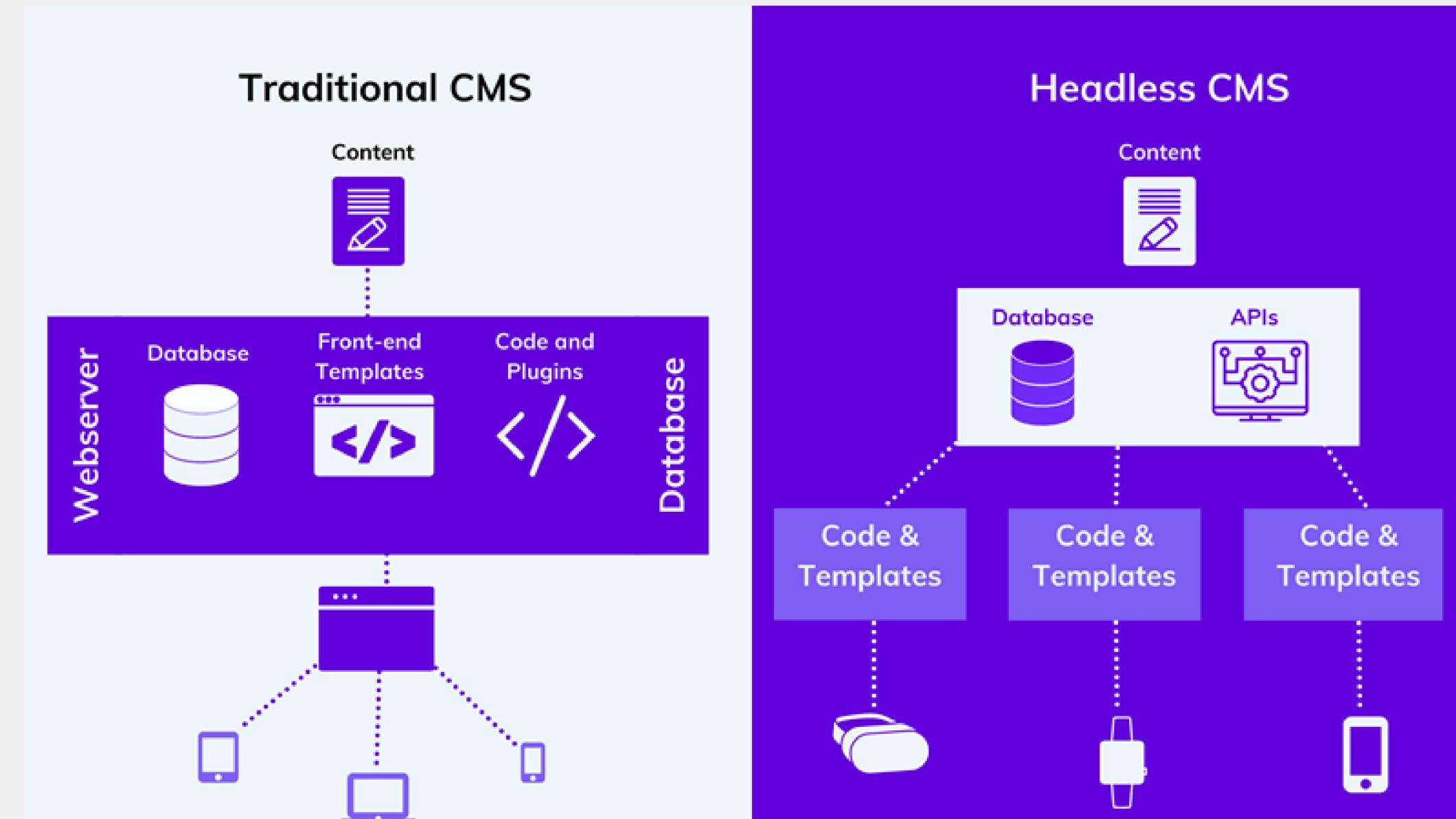
Monolith vs Headless



WAS?

Ein Headless CMS ermöglicht ...

- eine flexible und ungebundene Content-Verwaltung
- Bereitstellung der Inhalte über APIs
- nahtlose Einbindung in unterschiedliche Front-End-Technologien

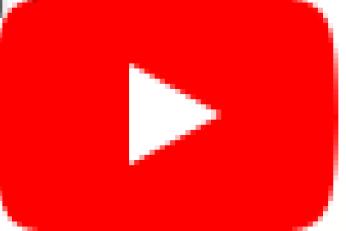


WAS?

 Headless CMS explained in 2 minutes

EXPLAINED IN 2 MINUTES

What is a headless CMS?



SANITY

Watch on  YouTube



Share 

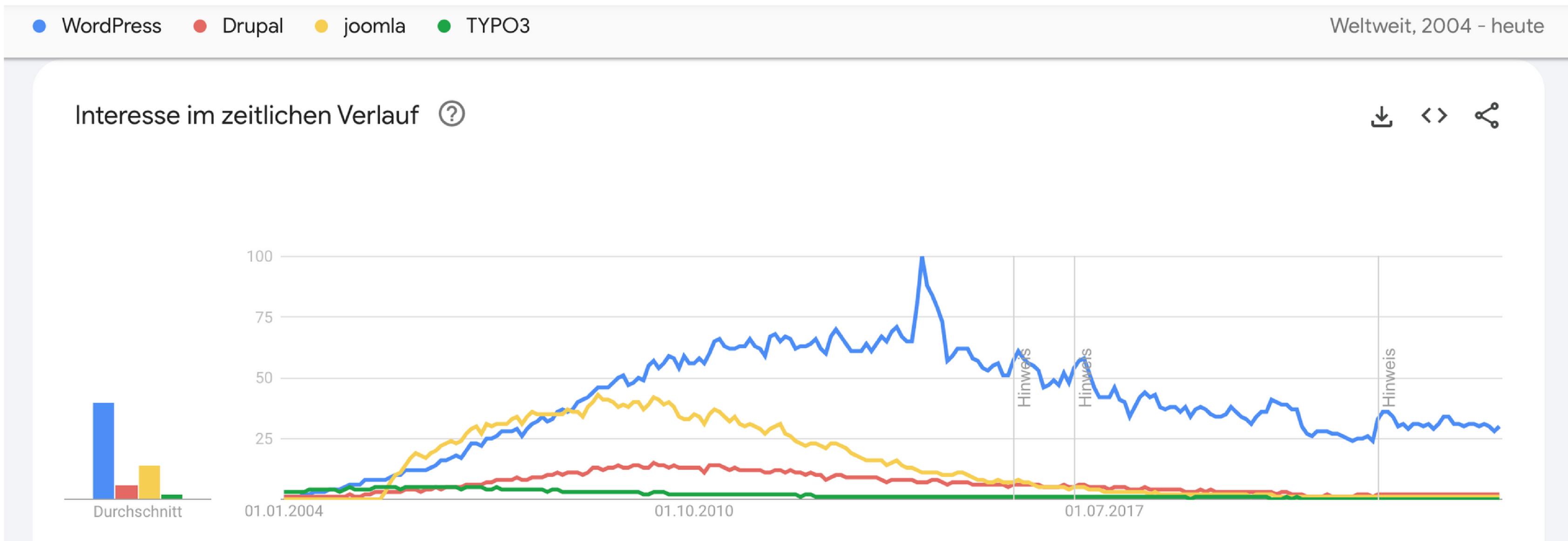
VERGLEICH

Monolith vs Headless

	Traditionelles CMS	Headless CMS
Platformunabhängigkeit	-	✓
Austauschbarer Technologie Stack	-	✓
Omnichannel Support	-	✓
API Support	-	✓
Plugin "Apokalypse"	✓	-
Lokalisierung & Mehrsprachigkeit	✓	✓
Code Wartbarkeit	-	✓

VERGLEICH

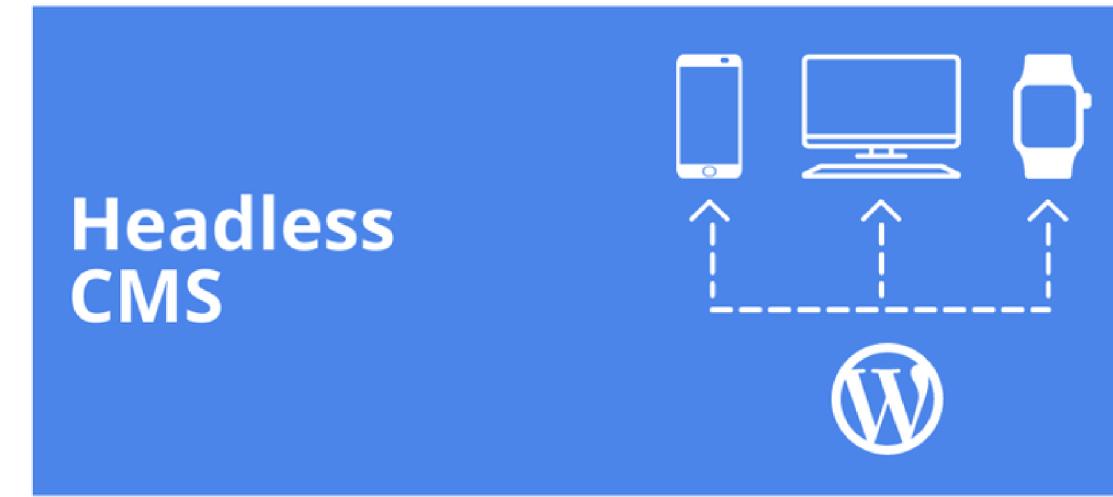
Die beliebtesten traditionellen CMS



Quellen: [Google Trends](#)

WORDPRESS

Decoupled CMS



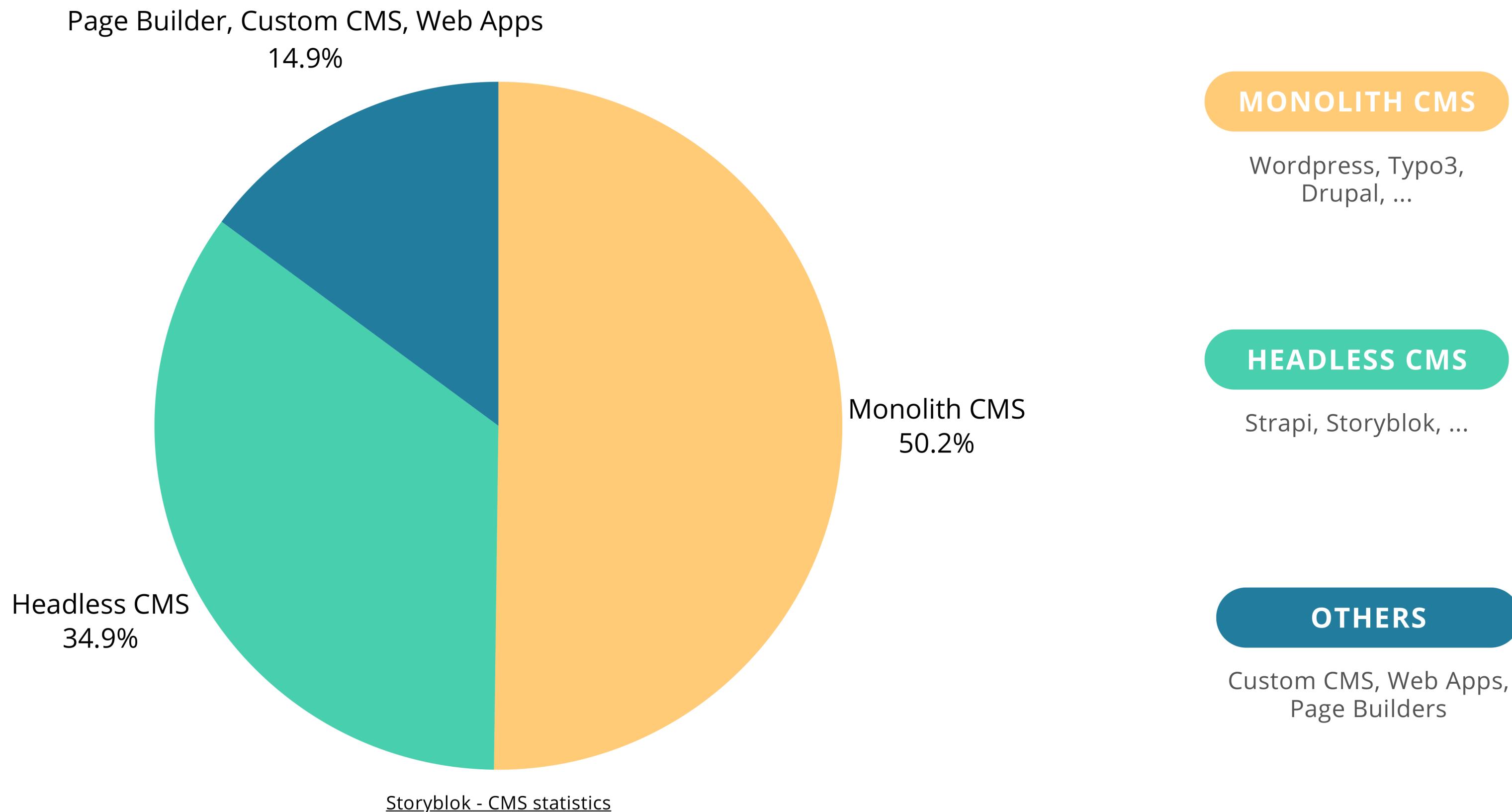
- Schnelle Backend erstellung (DEV)
- Gewohnte WordPress Backend für Content Creators
- Moderne Frontend Frameworks und SPA möglich (NPM, Webpack,...)
- ...

- Frontend Plugins nicht nutzbar
- Frontends von DEV abhängig
- SEO?
- In manchen Fällen ist Server Side Caching notwendig (z.B. mit Nuxt.js). Damit sind einige der genannten Vorteile wieder hinfällig.

[Artikel T3N: Headless CMS – warum wir WordPress in Zukunft ganz anders nutzen werden](#)

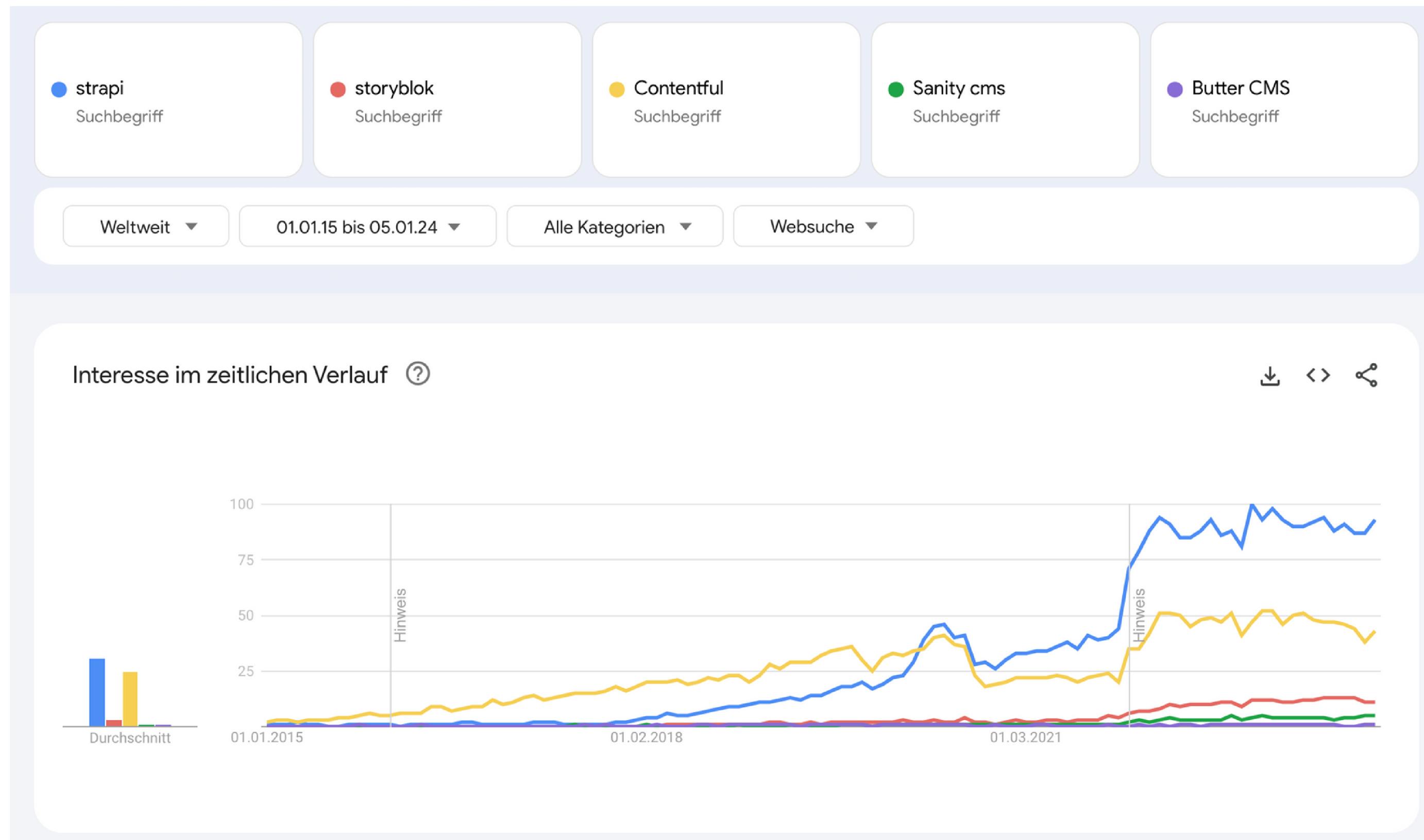
[Quelle](#)

VERBREITUNG



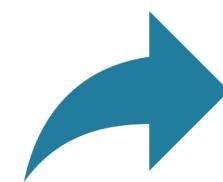
TRENDS

Most popular Headless CMS



TRENDS

Und in Österreich?



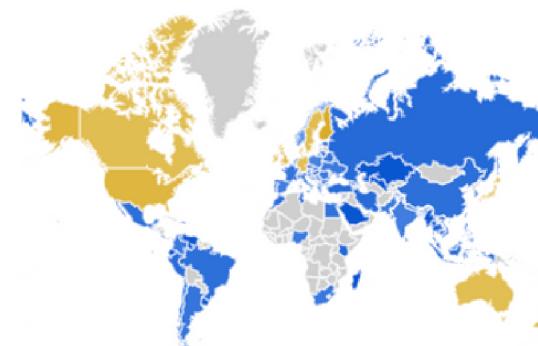
8,5-Millionen-Dollar-Investment für Linzer Start-up Storyblok (2021)

Aufschlüsselung nach Region

Region ▾

● strapi ● storyblok ● Contentful ● Sanity cms ● Butter CMS

Sortieren: Interesse an "storyblok" ▾



1	Österreich	
2	Kenia	
3	Irland	
4	Niederlande	
5	Schweden	

Die Farbintensität gibt die Prozentzahl der Suchanfragen an [WEITERE INFORMATIONEN](#)

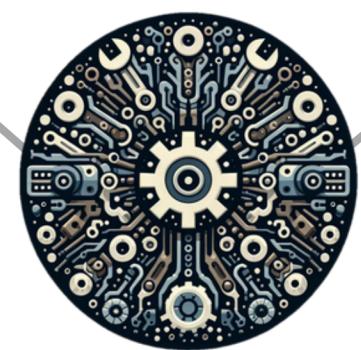
Regionen mit geringem Suchvolumen einschließen

◀ 1–5 von 82 Regionen werden angezeigt ▶

VORTEILE

Cloud storage

Einfache Zugänglichkeit auf unterschiedlichen Plattformen. Kein eigenes Hosting notwendig



Flexibilität

Headless CMS ermöglicht flexible Front-End-Gestaltung durch Trennung von Content und Präsentation.

Security

Gängige Angriffe wie DDOS-Angriffe (Denial of Service) kommen bei Headless CMS nicht vor.

Wartbarkeit

Bessere Versionierung und Update Policy als bei traditionellen

NACHTEILE

Einrichtung

Die Entwicklung eines Front-Ends erfordert spezielle Kenntnisse und kann komplexer sein als die Verwendung eines integrierten Front-Ends in traditionellen CMS.



Standardisierung

Da es keine festen Standards für die Umsetzung von Headless CMS gibt, kann die Integration mit verschiedenen Front-End-Technologien herausfordernd sein.



Agilität

Unabhängige Arbeit von Entwicklungs- und Content-Teams für schnellere Iterationen.



Komplexität

Nicht selbsterklärend für Content-Creator

Dev notwendig

Der erfolgreiche Einsatz eines Headless CMS hängt stark von den Fähigkeiten der Entwickler ab, da diese für die Gestaltung und Implementierung des Front-Ends verantwortlich sind.

Funktionalität

Traditionelle CMS bieten oft integrierte Funktionen für bestimmte Anwendungen (z.B., Blogs). Beim Einsatz eines Headless CMS müssen solche Funktionen möglicherweise selbst implementiert werden.

FAZIT

Wann setzt man ein Headless CMS ein?



Headless CMS ideal bei, ...

- ... **multiplen Plattformen**
- ... **kreativen & reaktiven Front-Ends**
- ... **agiler Entwicklung**
- ... **Abhängigkeit** von Content-Teams durch Tech-Teams
- ... Bedarf an **zukunftsorientierten, flexiblen Lösungen**

FAZIT

Wann setzt man auf ein traditionelles CMS?



Einsatz von traditionellem CMS wenn, ...

- ... **begrenzte Entwicklerressourcen** vorhanden sind
- ... **einfacher Webseite & nur eine Plattform**
- ... spezifische integrierte **CMS-Funktionen notwendig** sind (Blog, ...)

FAZIT

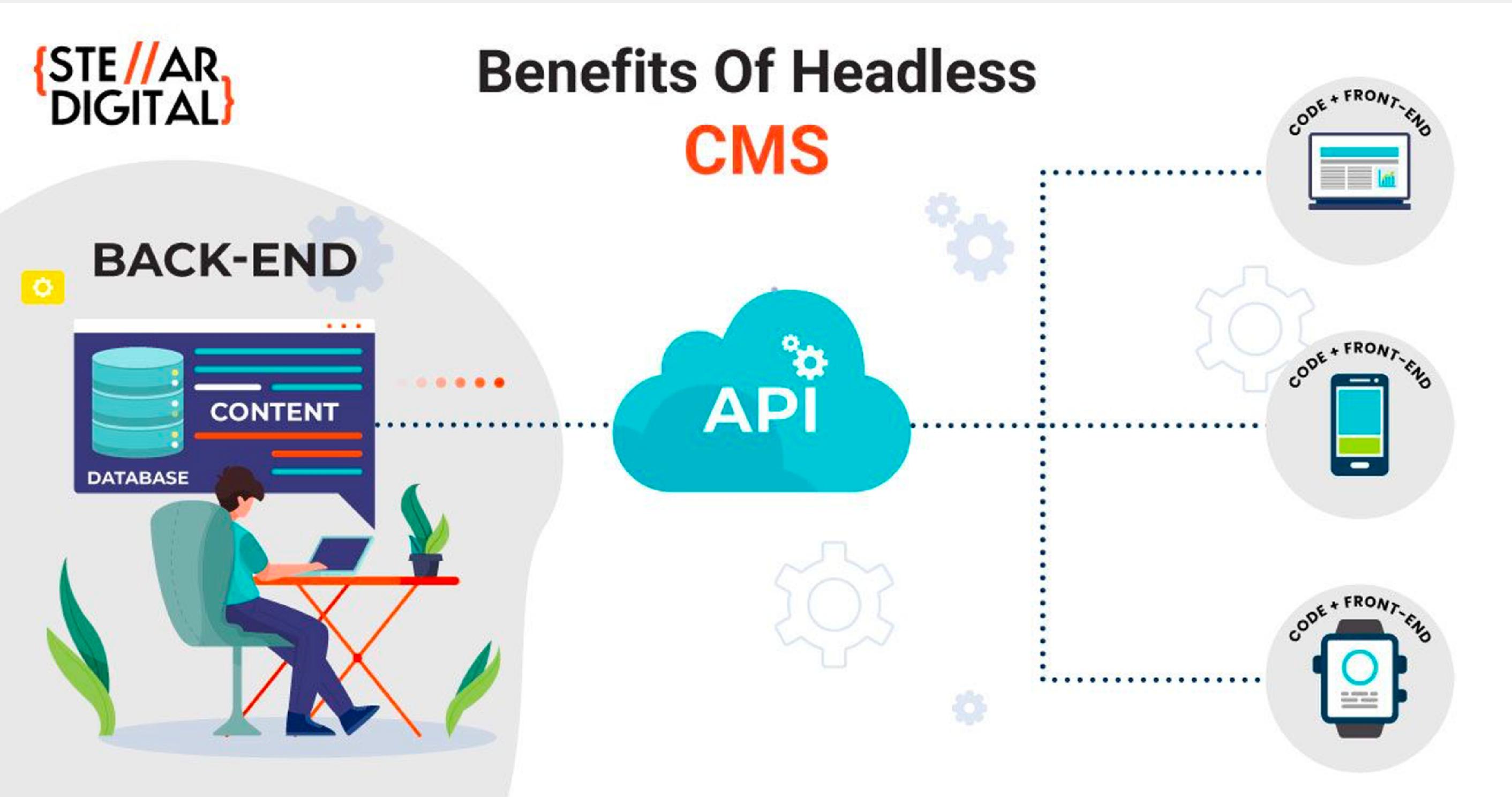
Headless CMS Traditional CMS Comparison Cheat Sheet

Feature	Headless CMS	Traditional CMS
Architecture	Modular	Monolithic
Content delivery	Omnichannel by nature	Content must be manually re-created for each channel
Scalability	Inherent to its architecture	Not easy to scale once built
Integrations	Microservices are easy to add and remove	Only via plugins
Security	Average security level	Plugins may increase security threats
Usability	Requires some tech-savvy	Often easier for low-tech users
Implementation	Requires a tech team	An option for teams with no tech support
Hosting	In the cloud	On an in-house server
Maintenance	Handled by vendor's team, delivered via the cloud	Handled in-house
Payment	Often subscription based	Usually all upfront



Sources: [headless cms vs monolith cms](#)

WORKSHOP



WORKSHOP

Wahl des CMS

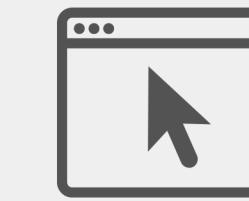
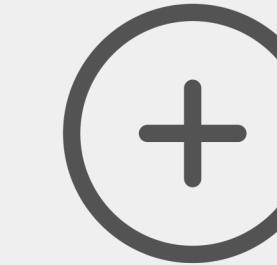


General / Pricing	Community	Entry	Business
	Free	99,00 €	849,00 €
Get Started for Free		Try for Free	Try for Free
Spaces (Project) included	1	1	1
Users/Seats included	1	5	20
Costs per additional seat (in one space)	9,00 €	9,00 €	9,00 €
Users/Seats Maximum	max 10	max 30	max 40
Monthly Traffic included	250GB	500GB	1TB
Uptime SLA	---	95%	97%

COMMUNITY	ENTERPRISE
Free, forever	Flexible pricing
SELF-HOSTED	SELF-HOSTED
Get Started	Contact sales
COMMUNITY	ENTERPRISE
<ul style="list-style-type: none">✓ Open-Source under MIT License✓ Role Based Access Control✓ REST & GraphQL API✓ Powerful Content Management✓ Infinite Customization Capabilities✓ Unlimited Usage (Entries, API calls, etc.)✓ Community Support	<ul style="list-style-type: none">✓ Everything in Community, plus...✓ Audit Logs✓ SSO✓ Review workflows✓ Vercel Visual Editing✓ Technical support with SLAs✓ Customer Success Manager✓ Technical onboarding

WORKSHOP

Zwei unterschiedliche Plattformen



NuxtJS



ionic

Warum?

Sehr gute Dokumentation, weit verbreitet

[strapi-nuxt docs](#)

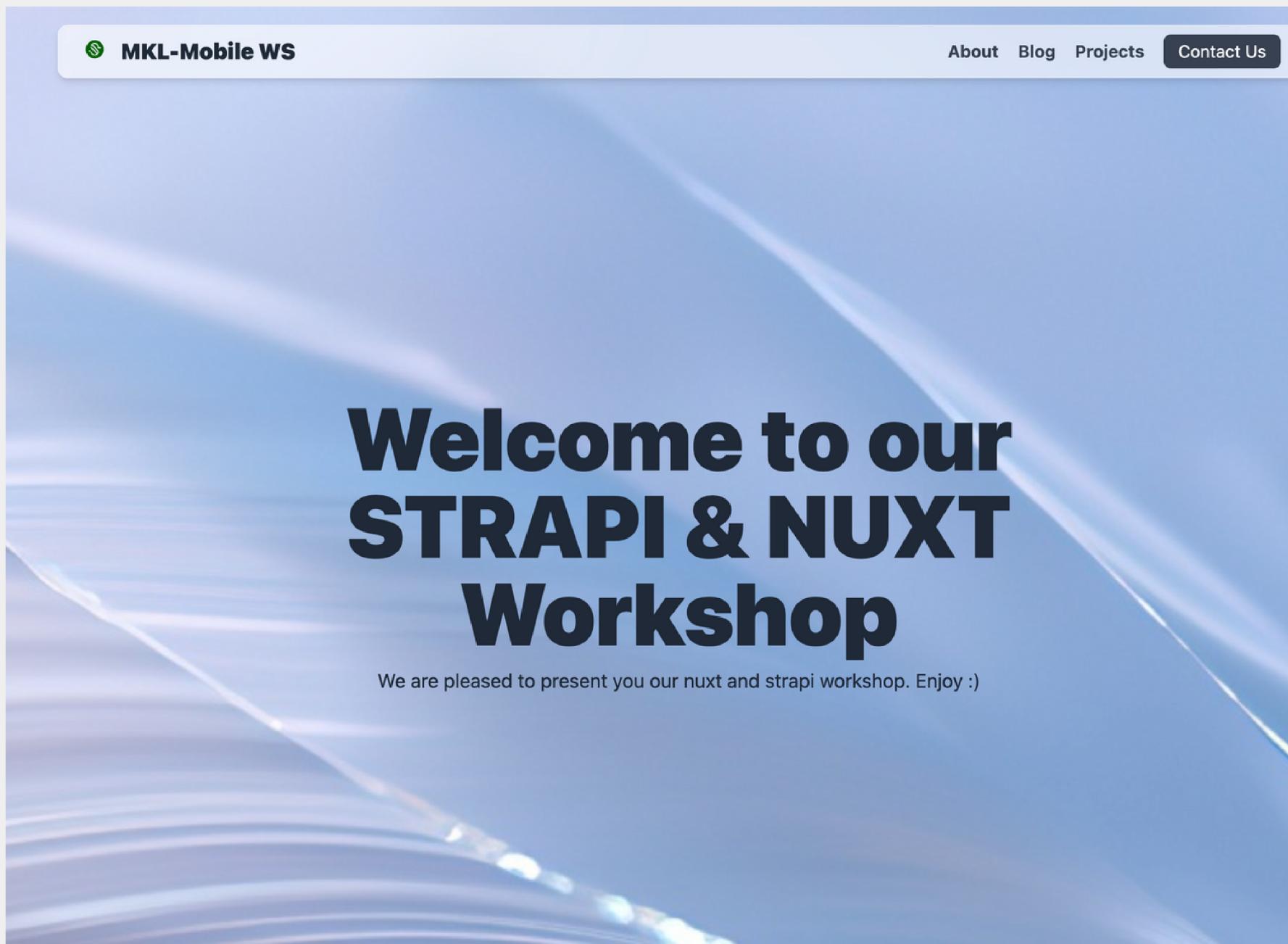
[build portfolio page](#)

[strapi - Ionic](#)

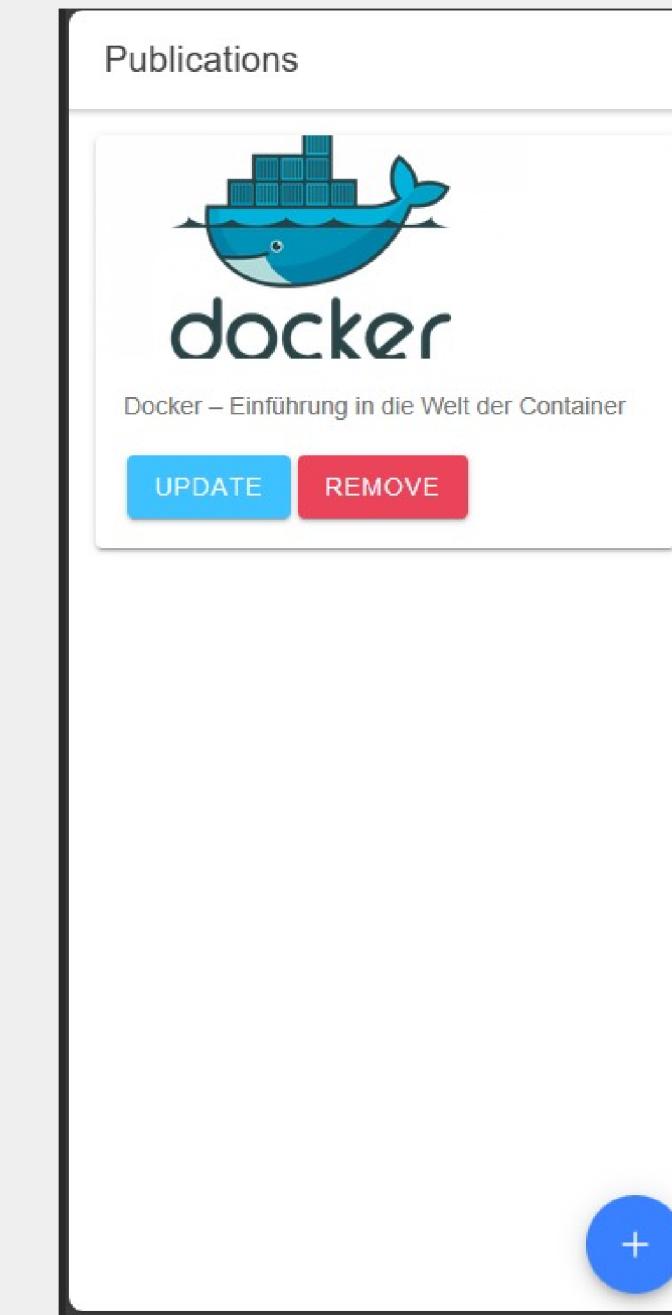
WORKSHOP

Was wir erstellen

Simble Portfolio Website (Nuxt)

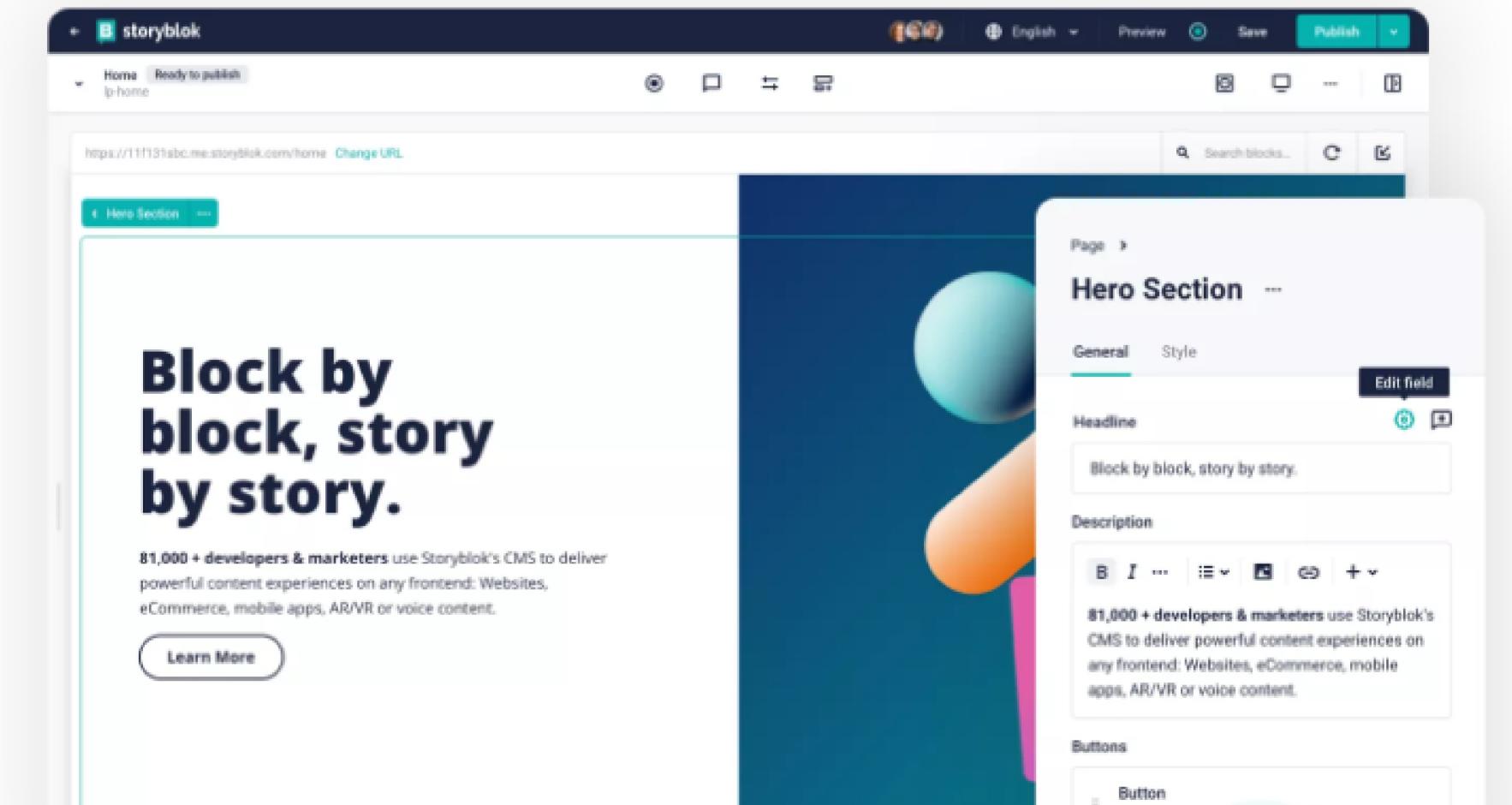


Mobile Web App (Ionic)



WORKSHOP

Storyblok



Storyblok vs strapi

THANK YOU



<https://github.com/fhstpoelten-avr21/strapi-workshop-frontend/tree/workshop>



Pusheen

