

# HEADLESS CMS

Überblick & Umsetzung

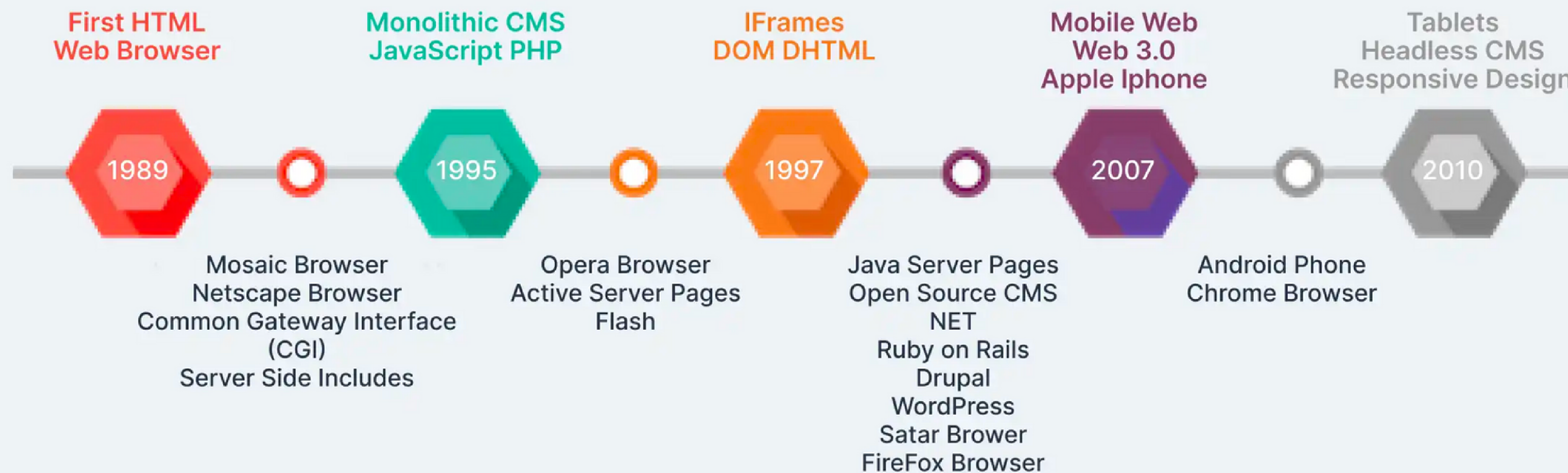
# AGENDA

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

# HISTORY

Entstehung des CMS im Web

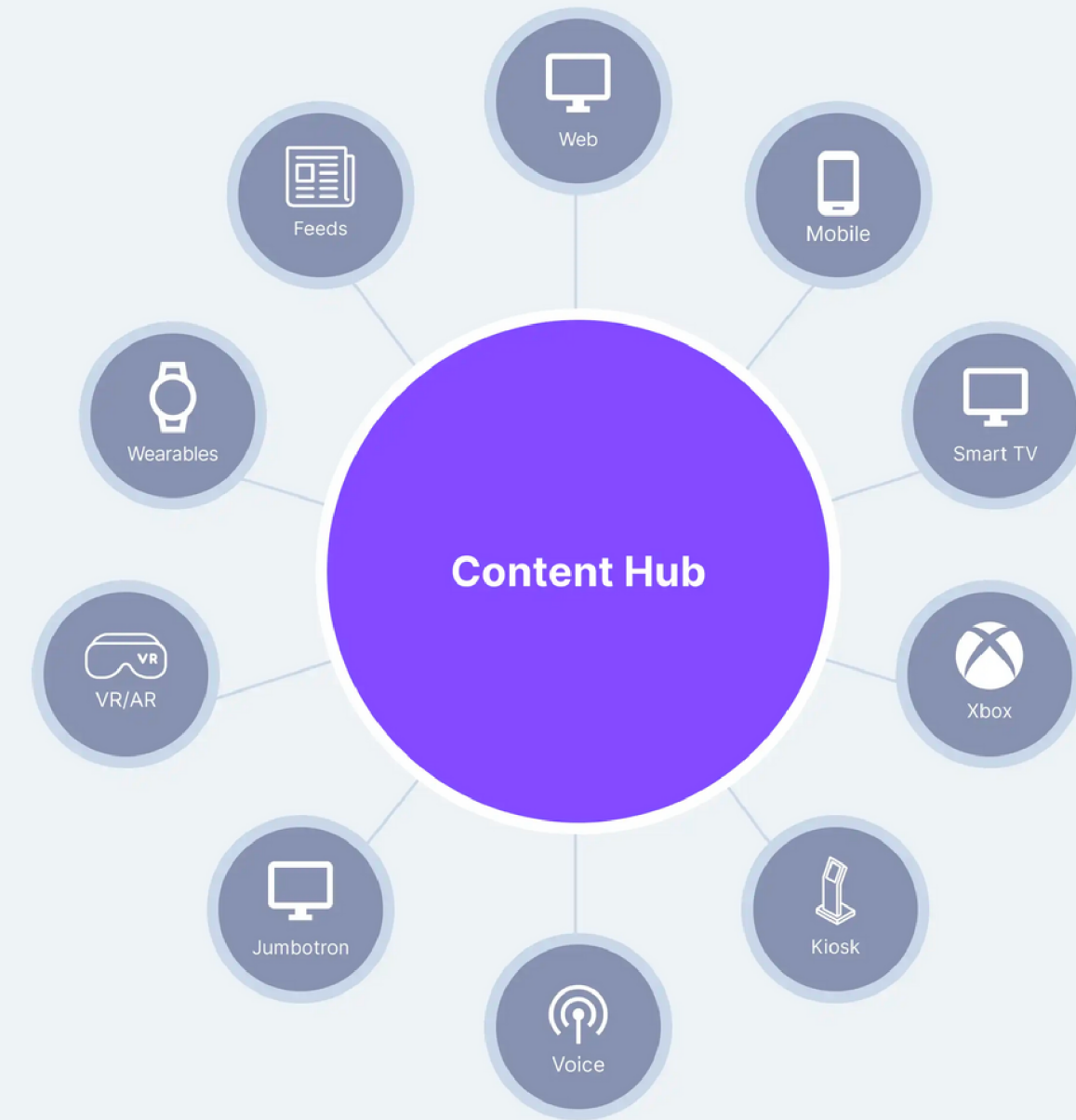
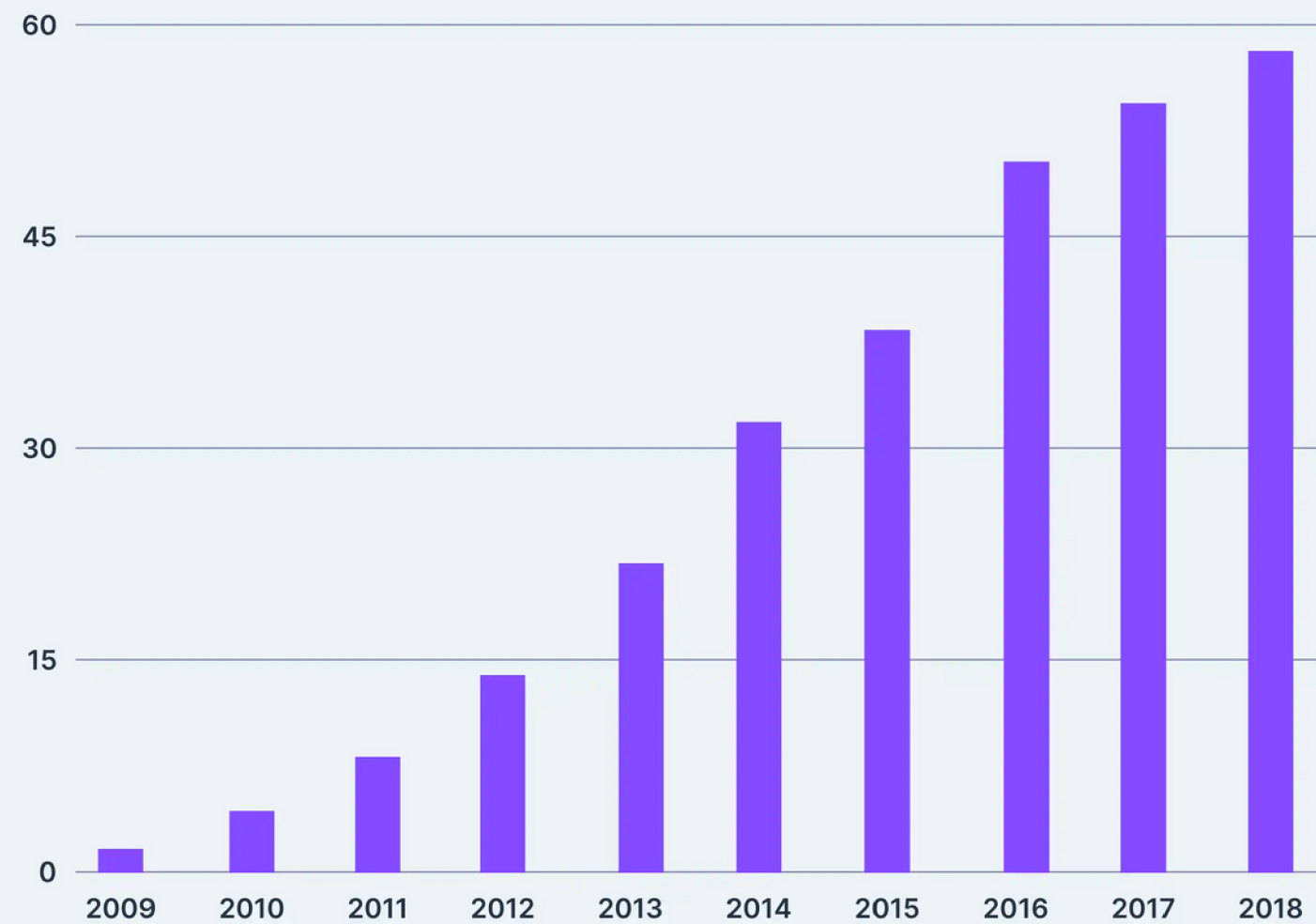
## Milestones for web content management



# HISTORY

## Vergleich traditionale CMS

Mobile Growth - Market Share Percentage



[History of content management systems and rise of headless CMS;](http://gs.statcounter.com/)

StatCounter Global Stats (<http://gs.statcounter.com/>)

# 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

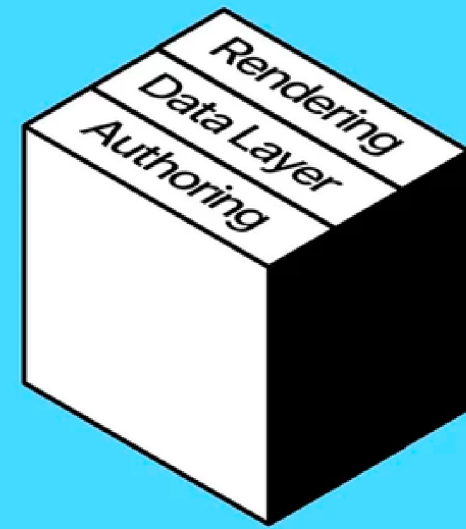
# WAS?

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.

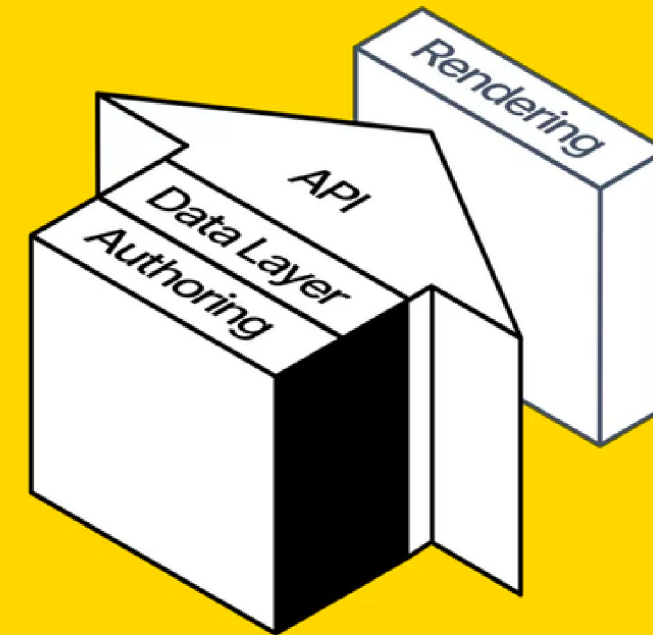
# VERGLEICH

## Monolith vs Headless



### Traditional CMS

Monolithic  
Self-host  
Websites



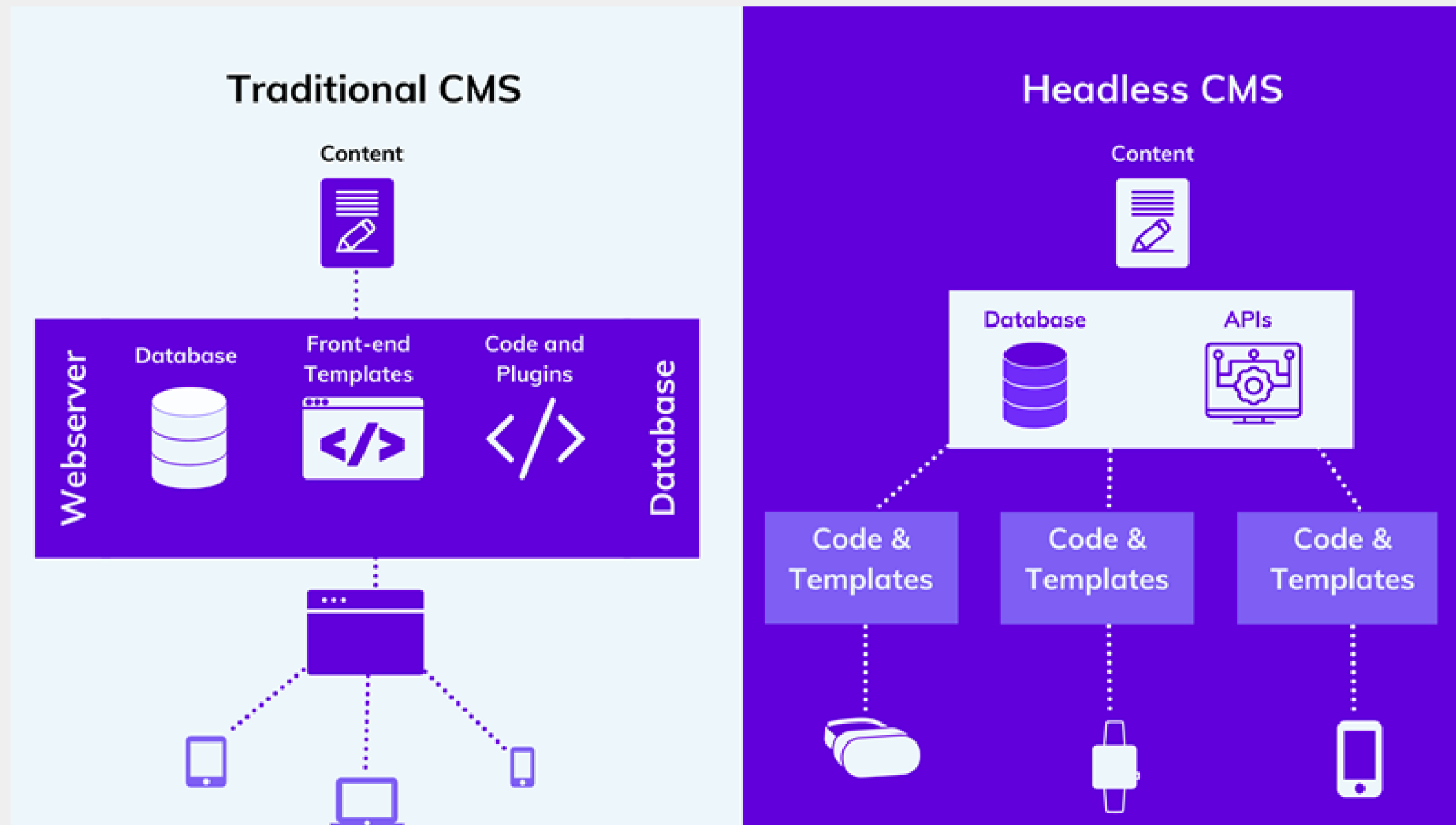
### Headless CMS

Semi-decoupled  
SaaS or Hosted  
Web Oriented

# 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

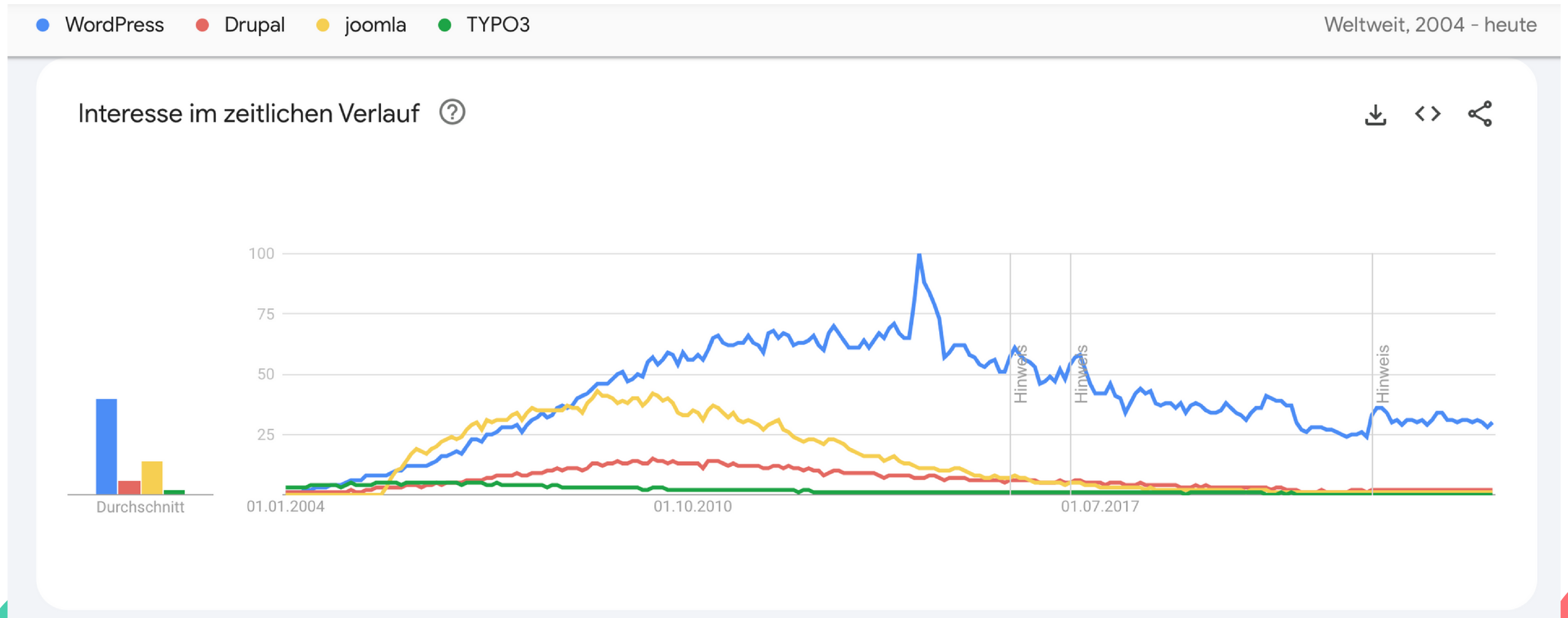
## Monolith vs Headless

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

Storyblok

# VERGLEICH

## Die beliebtesten traditionellen CMS



Quellen: [Google Trends](#)

# WORDPRESS

## Decoupled CMS



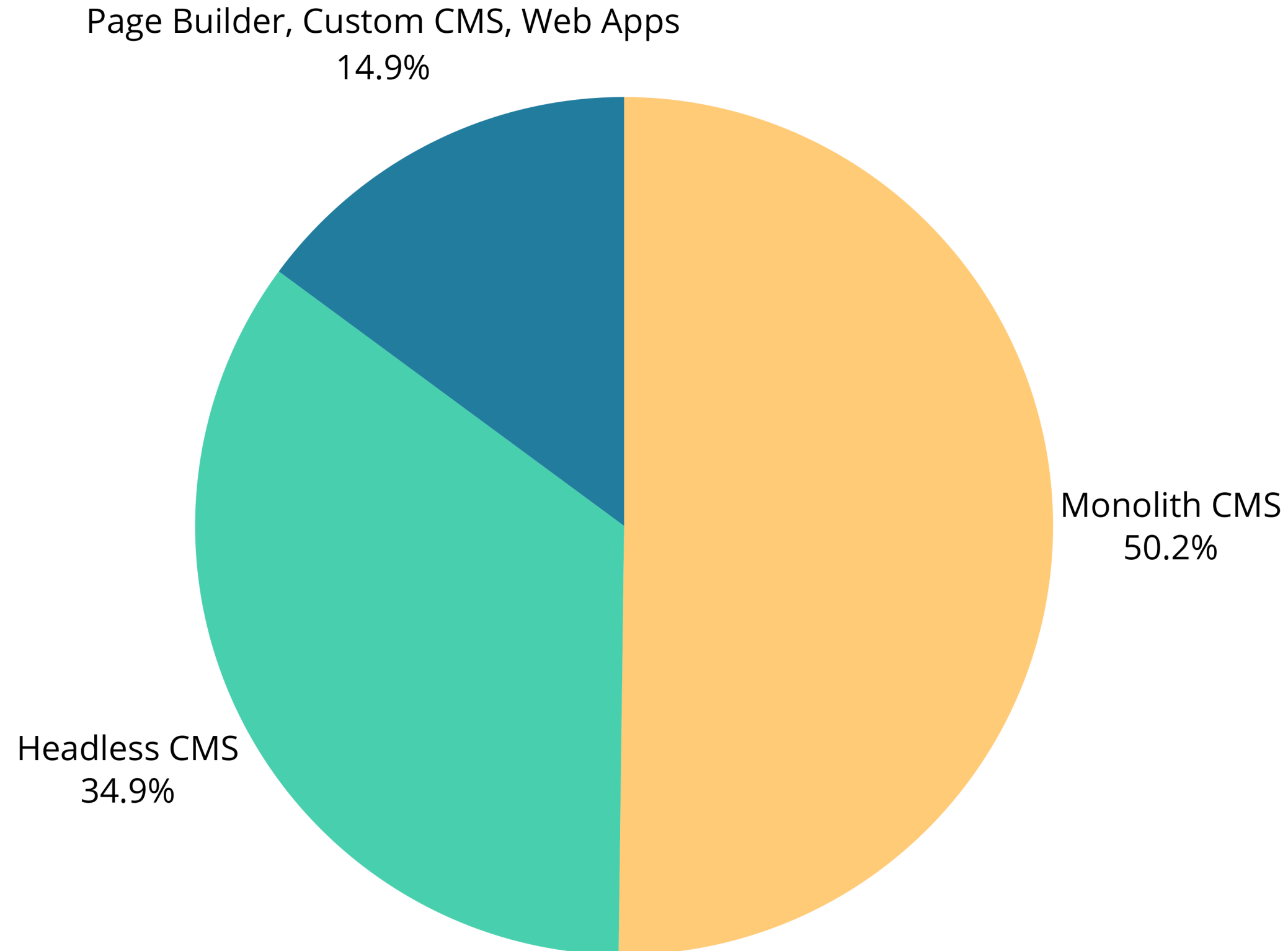
- Schnelle Backend erstellung (DEV)
- Gewohnte WordPress Backend für Content Creators
- Moderne Frontend Frameworks und SPA möglich (NPM, Webpack,...)
- ...

- 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



## MONOLITH CMS

Wordpress, Typo3,  
Drupal, ...

## HEADLESS CMS

Strapi, Storyblok, ...

## OTHERS

Custom CMS, Web Apps,  
Page Builders

# TRENDS

## Most popular Headless CMS

● **strapi**  
Suchbegriff

● **storyblok**  
Suchbegriff

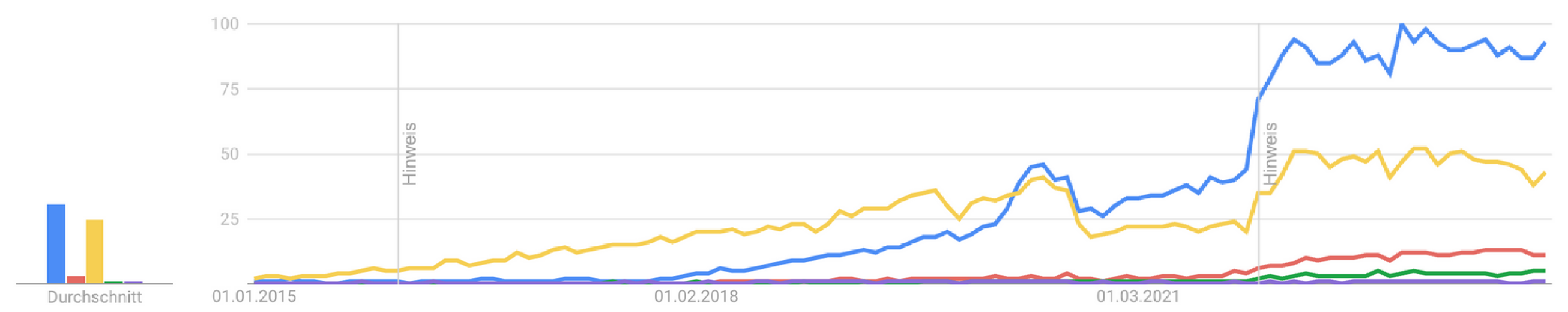
● **Contentful**  
Suchbegriff

● **Sanity cms**  
Suchbegriff

● **Butter CMS**  
Suchbegriff

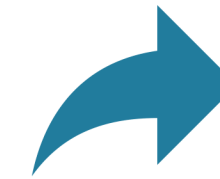
Weltweit ▼ 01.01.15 bis 05.01.24 ▼ Alle Kategorien ▼ Websuche ▼

Interesse im zeitlichen Verlauf ⓘ ↓ <> ↻

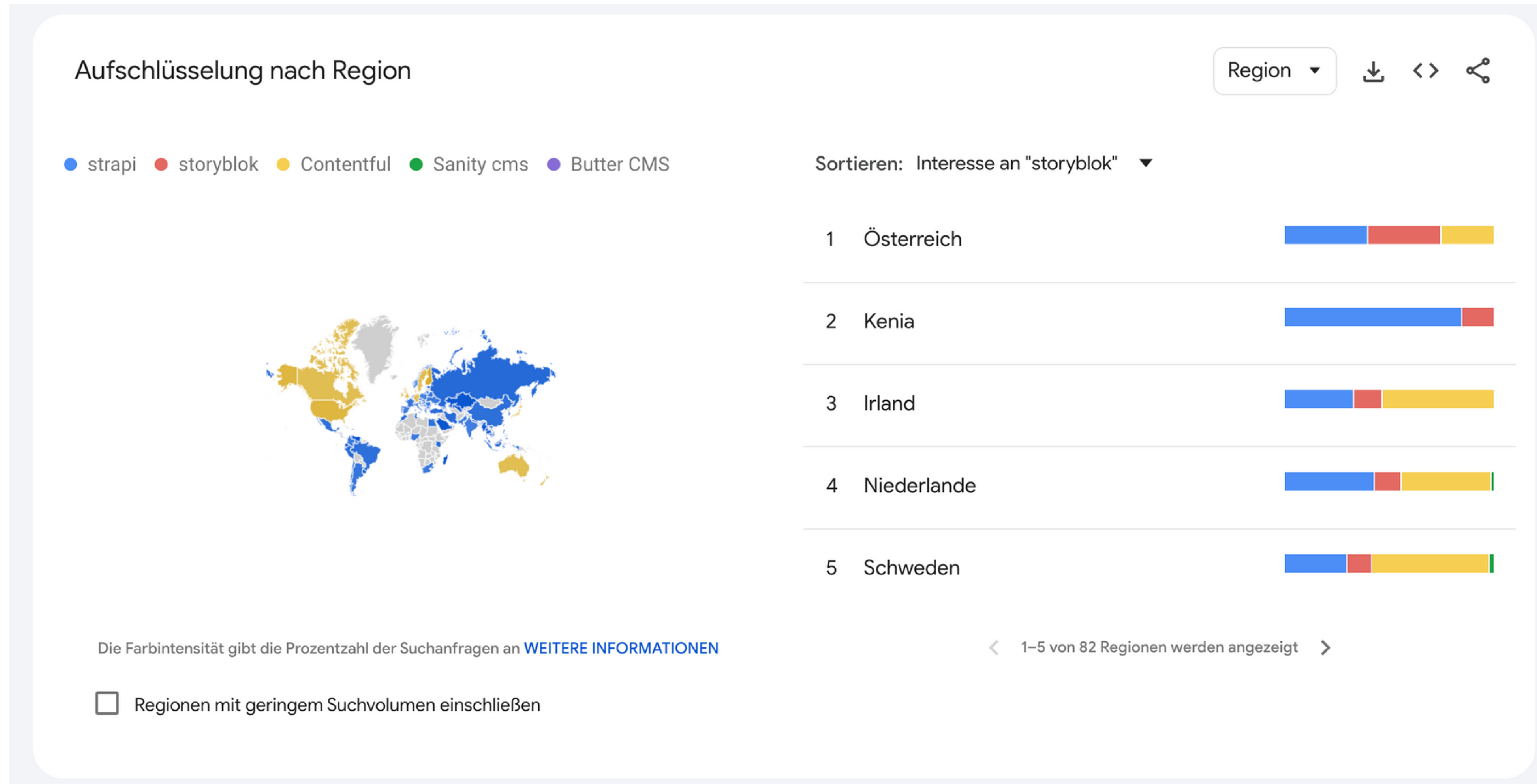


# TRENDS

Und in Österreich?



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



# VORTEILE

## Cloud storage

Einfache Zugänglichkeit auf unterschiedlichen Plattformen. Kein eigenes Hosting notwendig

## Dev friendly

Einfache Nutzung von APIs für Entwickler

## Agilität

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



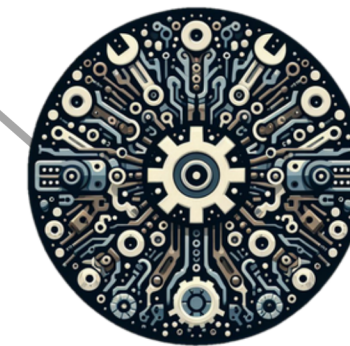
## Flexibilität

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.



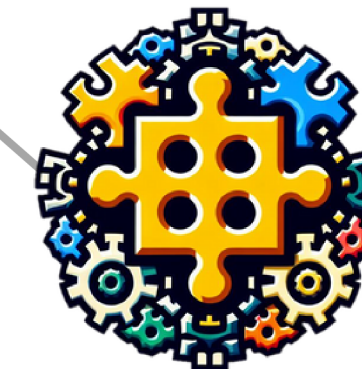
## Komplexität

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

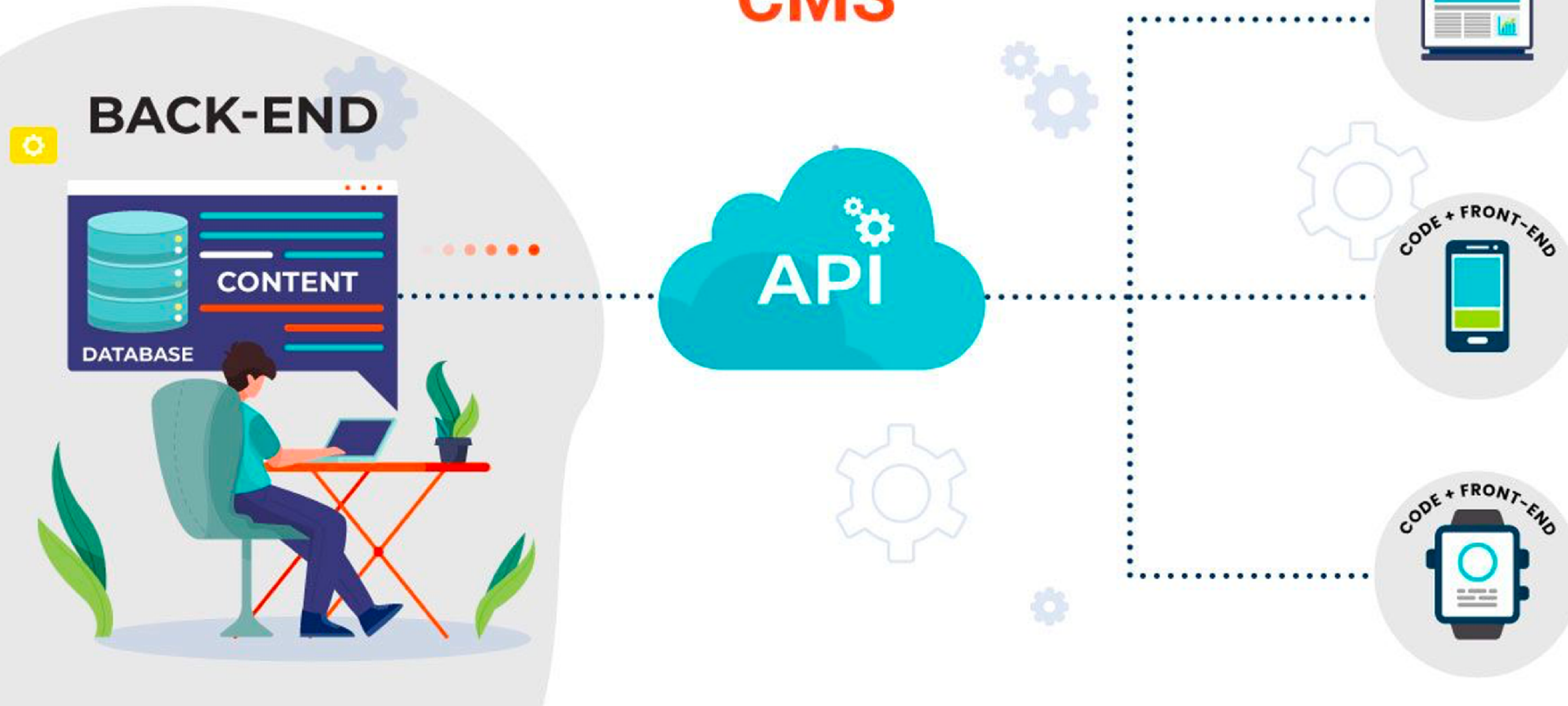


Sources: [headless cms vs monolith cms](#)

# WORKSHOP

{STE//AR  
DIGITAL}

## Benefits Of Headless CMS



# WORKSHOP

## Wahl des CMS



General / Pricing	Community	Entry	Business
<input type="checkbox"/> Only show differences	Free Get Started for Free	99,00 € Try for Free	849,00 € 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

**Free, forever**

SELF-HOSTED

[Get Started](#)

- ✓ 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

### ENTERPRISE

**Flexible pricing**

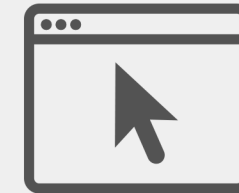
SELF-HOSTED

[Contact sales](#)

- ✓ 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?

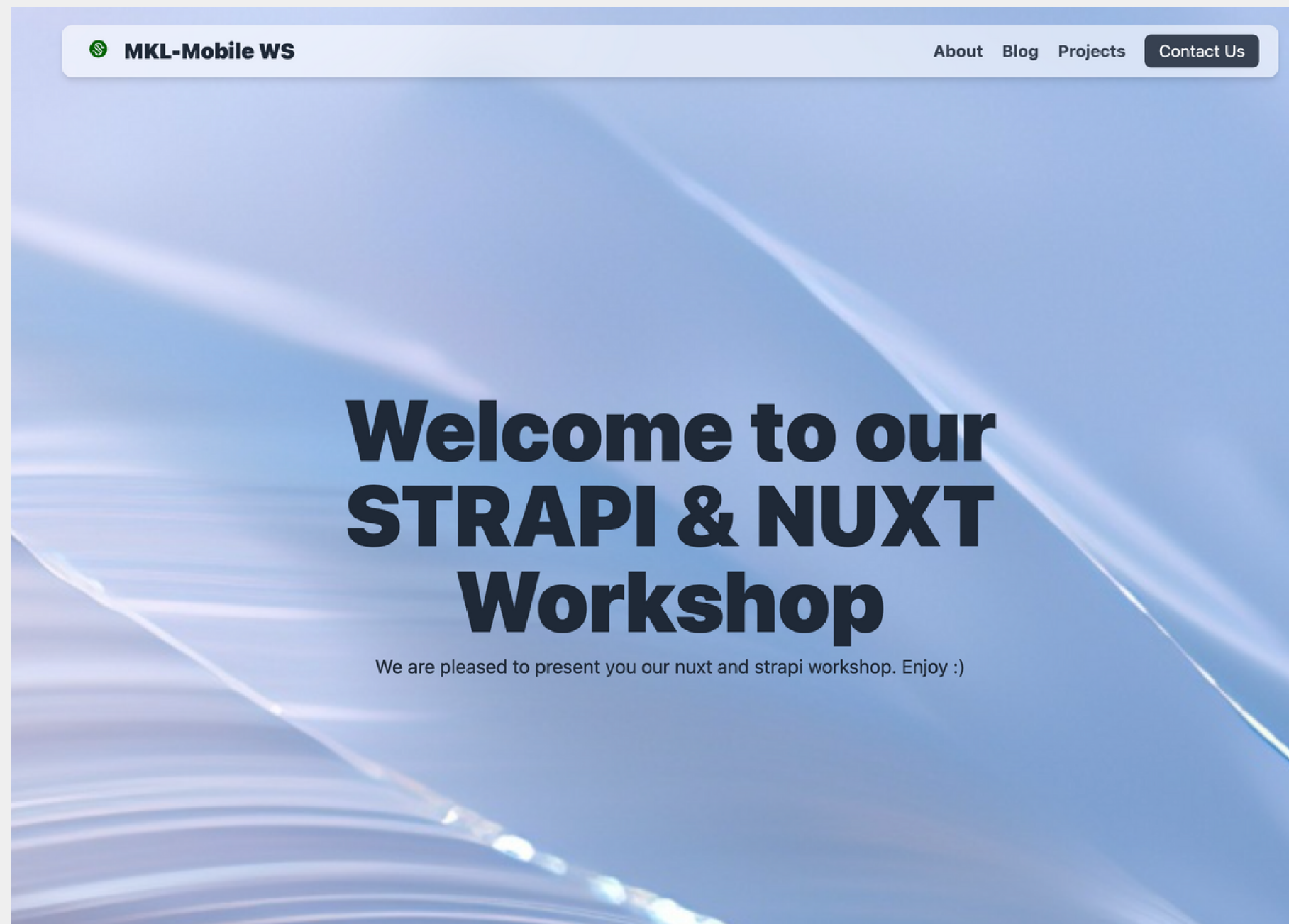
Sehr gute Dokumentation, weit verbreitet

[strapi-nuxt docs](#) [build portfolio page](#) [strapi - ionic](#)

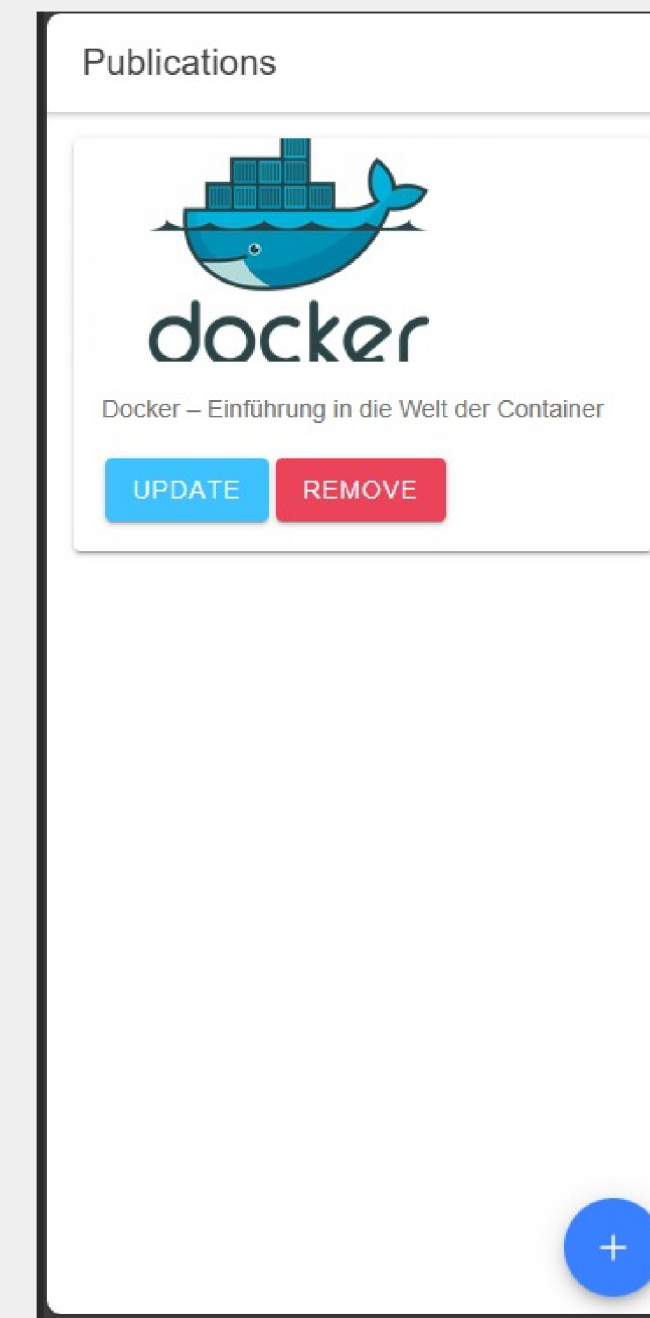
# WORKSHOP

Was wir erstellen

## Simple Portfolio Website (Nuxt)



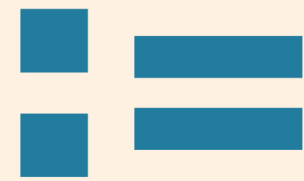
## Mobile Web App (Ionic)







# THANK YOU



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



Pushheen

