

ZUKUNFT IM E-COMMERCE

Headless, API-First & Co.



Landsiedel

0

Headless, API-First & Co.
Einführung

Headless, API-First & Co.

Einführung

- **Explosives Wachstum und große Chancen:** Gesamtumsatz E-Commerce 2007 bei knapp 136 Mrd. US-Dollar, 2022 bei 5,7 Billionen US-Dollar
- **Mobile Revolution:** Etwa 73% der Internetnutzer werden bis 2025 ausschließlich über Mobilgeräte auf das Internet zugreifen. Daher sind flexible und adaptive E-Commerce Plattformen essenziell
- **Mehr als nur ein Shop:** 55% der Online-Käufer sagen, dass sie lieber in einem Online-Shop einkaufen, die personalisierte Einkaufserlebnisse bieten
- Dadurch musste sich auch die **Technologie** weiterentwickeln

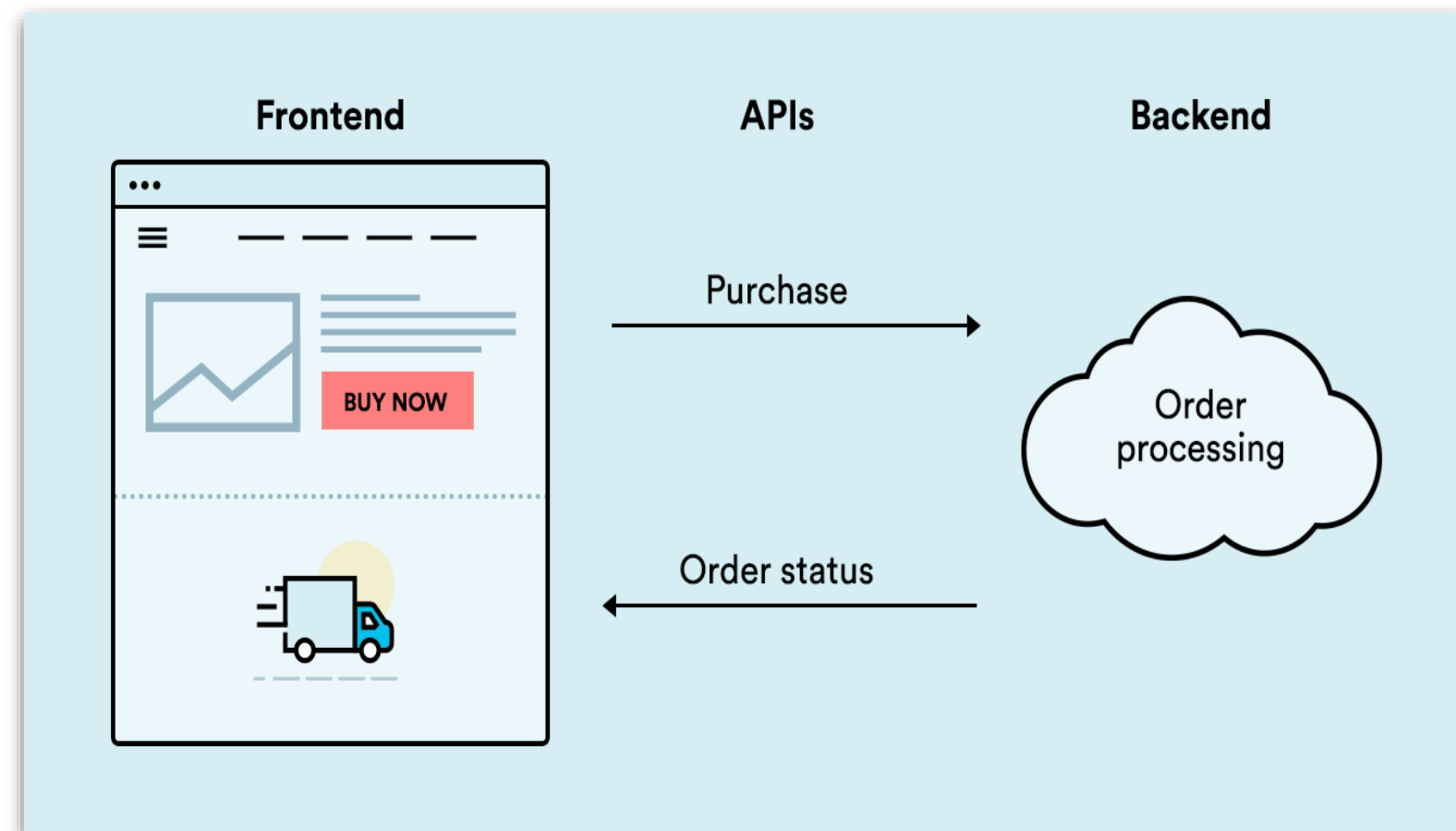
1

Headless, API-First & Co.
Definition Headless

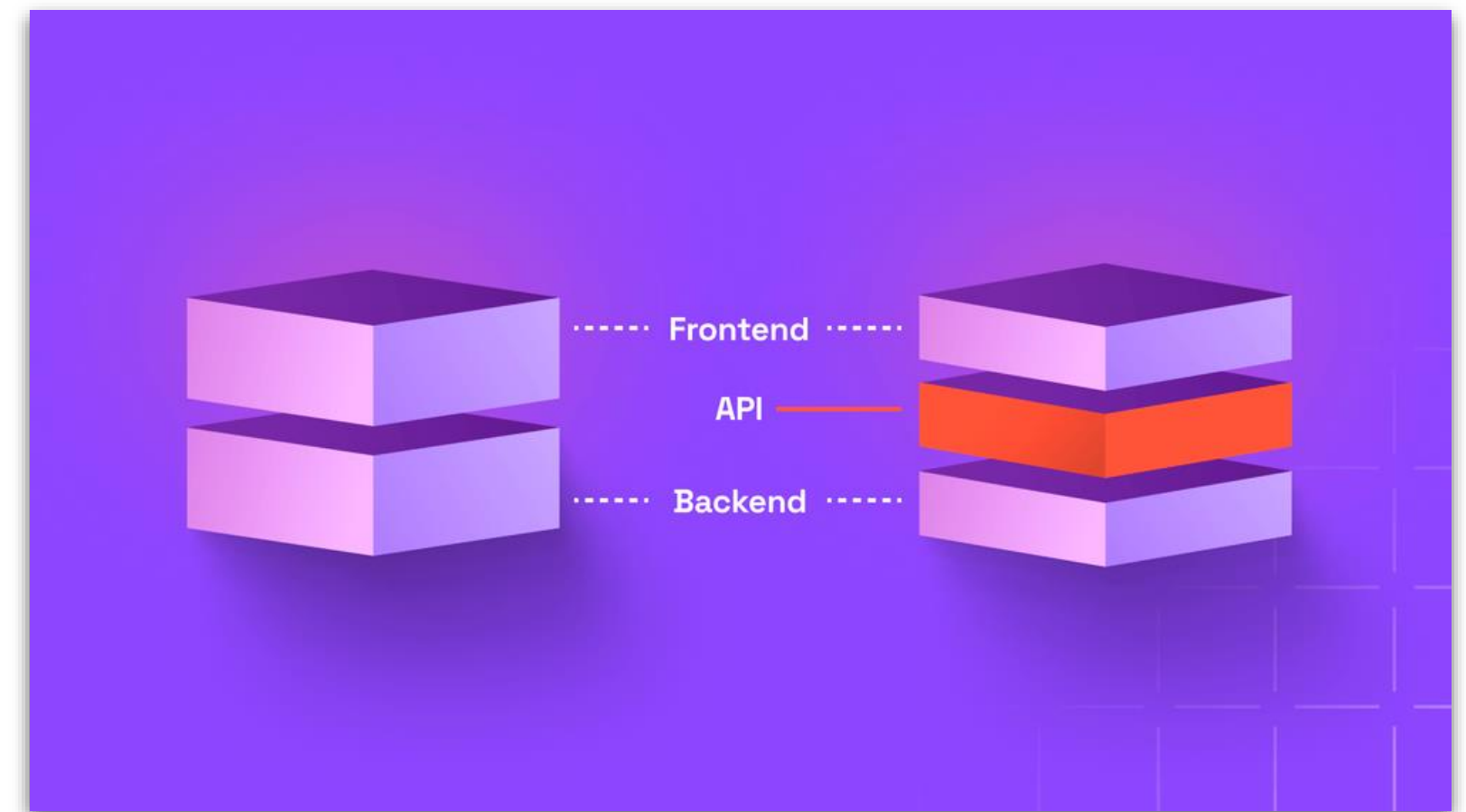
Headless bezeichnet Systeme, die ohne ein festes Frontend (oder "Head") betrieben werden. Das Backend und das Frontend sind entkoppelt. Dadurch kann das Frontend unabhängig vom Backend entwickelt und betrieben werden.

Headless, API-First & Co.

Definition Headless



Ablauf einer Bestellung



Klassische Struktur vs. Headless

Headless, API-First & Co.

Definition Headless - Bedeutung für das Ökosystem

- **Mehrere Frontends:** Ein Backend kann verschiedene Frontends bedienen (z.B. Web, Mobile, IoT)
- **Schnelle Anpassung:** Frontend-Updates ohne Backend-Änderungen sind möglich
- **Geräteübergreifende Konsistenz:** Einheitliche Datenverarbeitung für verschiedene Geräte und Plattformen

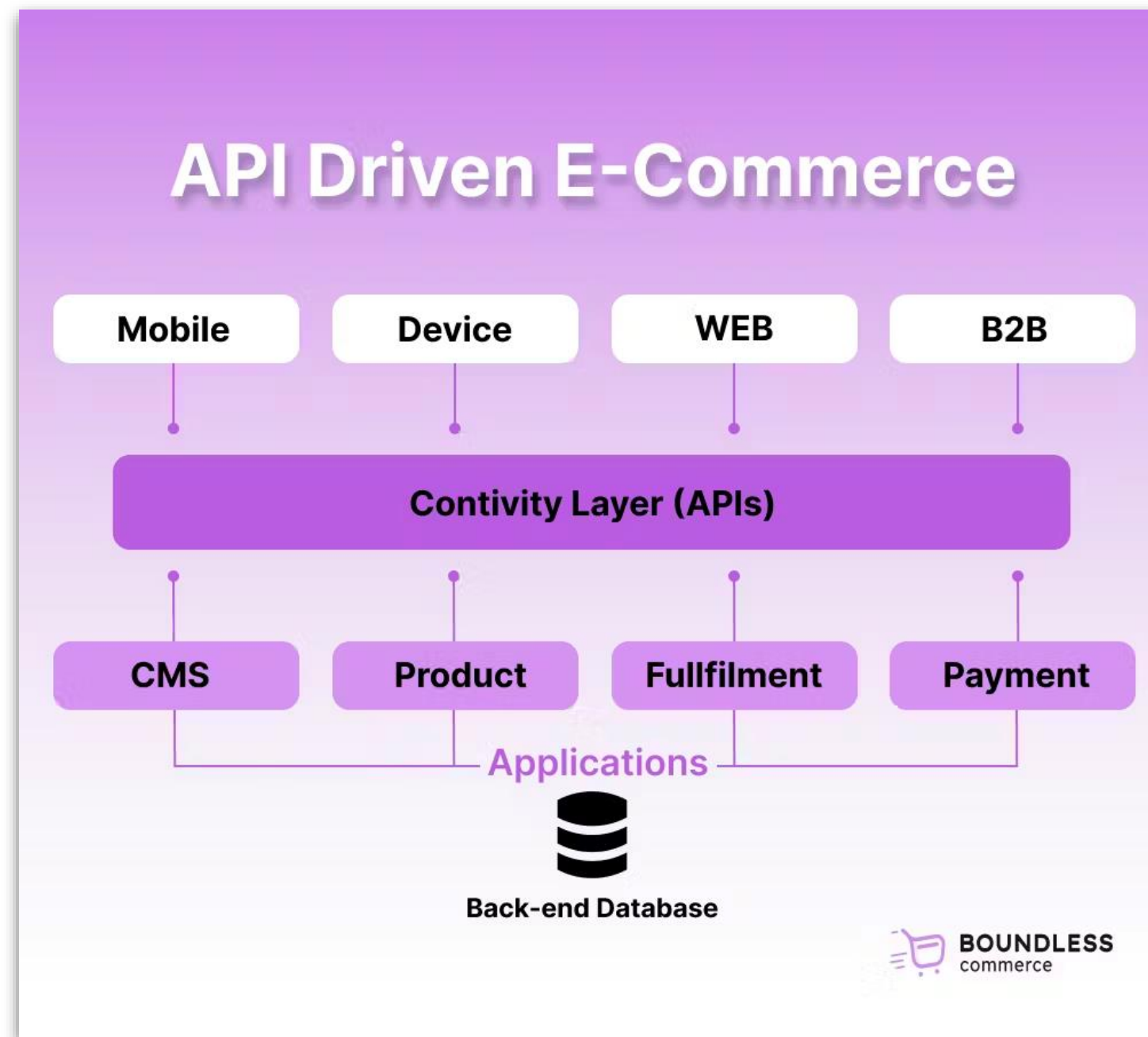
2

Headless, API-First & Co.
Definition API-First

Beim **API-First Ansatz** wird zuerst die API (Schnittstelle) entwickelt, bevor andere Teile der Anwendung erstellt werden. Dies stellt sicher, dass die Schnittstelle konsistent und robust ist und von verschiedenen Frontends oder Systemen verwendet werden kann.

Headless, API-First & Co.

Definition API-First



- Über die API werden Front- und Backend verbunden
- Verschiedene Kanäle und Frontends können ohne viel Aufwand angebunden werden

Headless, API-First & Co.

Definition API-First - Bedeutung für das Ökosystem

- **Standardisierte Kommunikation:** Ermöglicht die Kommunikation zwischen verschiedenen Anwendungen und Plattformen
- **Schnellere Integration:** Vereinfacht das Hinzufügen von Funktionen oder Partnerschaften
- **Skalierbarkeit:** Erweitert oder verändert Anwendungen ohne große Umbauten

3

Headless, API-First & Co.
Technische Anforderungen

Grundlegende Server-Anforderungen

Eine Headless-Architektur erfordert robuste und leistungsfähige Server, die in der Lage sind, API-Anfragen effizient zu verarbeiten und schnell zu antworten.

A large orange shape, resembling a quarter-circle or a stylized '0', occupies the bottom right corner of the slide. Inside this shape, the number '01' is written in a large, white, sans-serif font.

01

Cloud vs. On Premise

Cloud-Lösungen bieten Skalierbarkeit und Flexibilität, ideal für Headless- und API-First-Ansätze, während traditionelles Hosting möglicherweise weniger flexibel ist, kann aber in einigen Fällen kostengünstiger sein.

03

Abhängig vom Umfang und den Anforderungen des Projekts können relationale Datenbanken wie MySQL oder nicht-relationale Datenbanken wie MongoDB zum Einsatz kommen.

Datenbanken

Während Headless-CMS oft in JavaScript (z.B. React, Vue.js) entwickelt werden, erfordert API-First eine Vielzahl von Entwicklungstools und Plattformen, um APIs zu erstellen und zu warten.

**Entwicklungs-
plattformen**

Headless, API-First & Co.

Technische Anforderungen - Integration in bestehende Systeme

- Middleware oder API-Connectoren sind essentiell, um nahtlos mit anderen Tools, Systemen und Plattformen (wie CRM, E-Commerce-Plattformen oder Datenbanken) zu kommunizieren
- Bei der Umstellung auf Headless- oder API-First-Systeme kann die Migration von Daten aus älteren Systemen eine Herausforderung darstellen
- Es ist wichtig zu überprüfen, ob bestehende Technologien und Tools mit dem neuen System kompatibel sind oder ob Anpassungen erforderlich sind

Headless, API-First & Co.

Technische Anforderungen - Sicherheitsanforderungen

- Bei der Übertragung von Daten zwischen Frontend und Backend sowie bei der Speicherung von Daten ist eine solide Verschlüsselung unerlässlich
- Authentifizierung und Autorisierung: Sicherstellen, dass nur berechtigte Benutzer auf bestimmte Daten oder Funktionen zugreifen können
- Monitoring und Angriffserkennung: Regelmäßige Überwachung der Systeme, um mögliche Anomalien oder Sicherheitsverletzungen frühzeitig zu erkennen und zu beheben

4

Headless, API-First & Co.

Welche Kosten entstehen werden

Initialkosten

Die Erstellung einer maßgeschneiderten Lösung oder Anpassung vorhandener Systeme an einen Headless- oder API-First-Ansatz kann beträchtliche Startkosten verursachen.

A large, solid orange curved shape that starts from the bottom right corner and curves upwards and to the left, partially framing the page number.

01

Spezialisierte Werkzeuge und Plattformen

Tools, die speziell für die Arbeit mit diesen Ansätzen entwickelt wurden, sind möglicherweise nicht immer günstig z.B. API-Management-Tools, Headless-CMS-Lösungen oder spezielle Entwicklungsplattformen.

03

Die Einarbeitung deines Teams in neue Technologien und Praktiken erfordert Investitionen in Schulungen, Kurse und Workshops.

Schulung und Weiterbildung

Während traditionelle Systeme oft eine einmalige Investition sind oder monatliche Kosten verursachen, können Headless- und API-First-Systeme regelmäßige Aktualisierungen und Wartungsarbeiten erfordern.

**Wartung und
Support**

Skalierung und Infrastruktur

Wenn dein System wächst und mehr Traffic verarbeitet, können zusätzliche Infrastrukturkosten und -anforderungen entstehen, insbesondere wenn Cloud-Lösungen oder Drittanbieterdienste genutzt werden.

Sicherheitsinvestitionen

Da Daten oft über APIs ausgetauscht werden, sind Investitionen in robuste Sicherheitsmaßnahmen und -tools entscheidend, um Datenlecks oder Angriffe zu verhindern.

5

Headless, API-First & Co.
Vor- und Nachteile Headless

Headless, API-First & Co.

Vor- und Nachteile Headless

Pros

Die Trennung von Front- und Backend führt zu mehr Flexibilität



Multichannel-Bereitstellung, ohne großartige Backend Anpassungen



Ohne ein traditionelles CMS kann das Frontend schneller geladen werden.



Zukunftssicher und einfachere Design- und Plattformwechsel



Cons



Die Implementierung kann oft teurer sein als die von traditionellen Systemen



Die Gesamtkosten eines Projekts können höher sein



Anlernzeit für Entwickler und Redakteure



Standard CMS Funktionen müssen evtl. manuell hinzugefügt werden

6

Headless, API-First & Co.
Vor- und Nachteile API-First

Headless, API-First & Co.

Vor- und Nachteile API-First

Pros

Einfache Integration mit anderen Diensten und Tools



API-First Systeme können leichter skaliert werden



Hinzufügen von Diensten, etc. ohne das Kernsystem hinzuzufügen



Zukunftssicher durch standardisierte APIs



Cons



Implementierung und Wartung kann technisch anspruchsvoller sein



Eventuelle Abhängigkeit von Drittanbieter APIs



Entwicklung und Wartung von robusten APIs kann teuer sein



Schlechte Performance durch schlecht optimierte APIs

7

Headless, API-First & Co.
Best Practices Headless

Headless und API-First

Best Practices Headless



Zielgerichtete Planung

Vor der Implementierung eines Headless-Ansatzes sollte eine klare Strategie und Planung erfolgen, um sicherzustellen, dass die Architektur den Geschäftsanforderungen entspricht.



Frontend / PWA Starter Packages

Nutze den Vorteil der Plattformunabhängigkeit und gestalte das Frontend so, dass es auf allen Geräten reibungslos funktioniert. Aber Vorsicht bei den Kosten!



Sicherheitsaspekt bedenken

Da bei einem Headless-Ansatz das Frontend vom Backend getrennt ist, sollten spezielle Sicherheitsmaßnahmen ergriffen werden, insbesondere bei der Datenübertragung.

8

Headless, API-First & Co.
Best Practices API-First

Headless und API-First

Best Practices API-First



Standardisierte API-Dokumentation

Halte die API-Dokumentation immer aktuell und nutze Tools wie beispielsweise Swagger, um sie für Entwickler leicht verständlich zu machen.



Sicherheitsprotokolle nutzen

Verwende Authentifizierung, Autorisierung und andere Sicherheitsprotokolle, um deine API vor unerwünschten Zugriffen zu schützen.



Rate Limiting und Caching

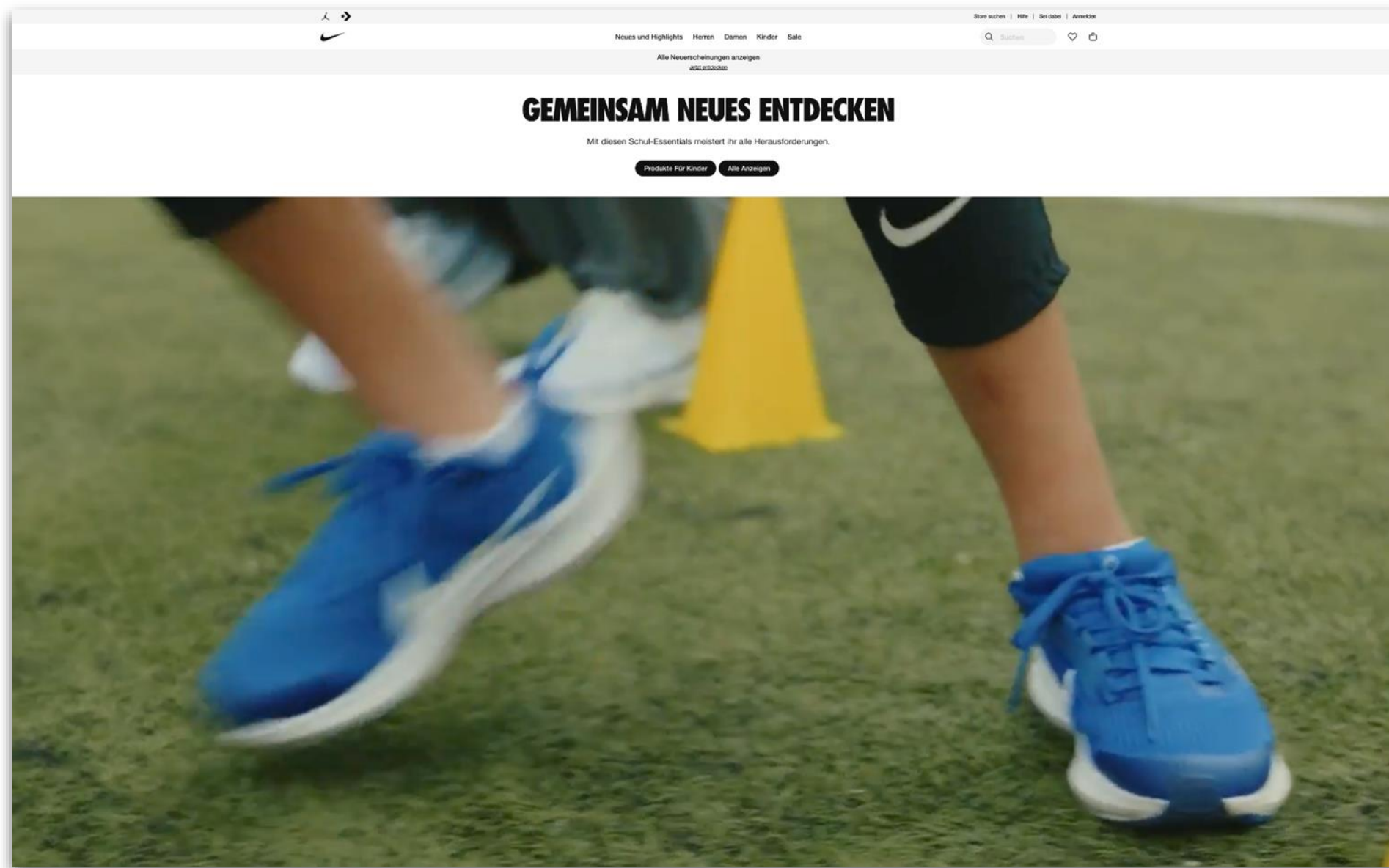
Implementiere Rate Limiting, um deine API vor Überbeanspruchung zu schützen, und nutze Caching, um die Performance zu optimieren.

9

Headless, API-First & Co.
Beispiel Headless

Headless und API-First

Beispiel Headless



- Wollen einen Mobile First Ansatz
- Passt am besten für die übergreifende Strategie
- Möchte damit den wachsenden Anforderungen gerecht werden
- Personalisierung, Kundenbeziehung stärken, Mobile-First

Headless und API-First

Beispiel Headless - Umsetzung

- Nikes Website und mobile App trennen das Frontend vom Backend
- Frontends können unabhängig voneinander angepasst werden
- Damit kann Nike die Kundenoberfläche schnell ändern
- Es sind keine Anpassungen im Backend notwendig
- Technologie: React SPA, kombiniert mit Node.js backend for frontend

Headless und API-First

Beispiel Headless - Ergebnisse

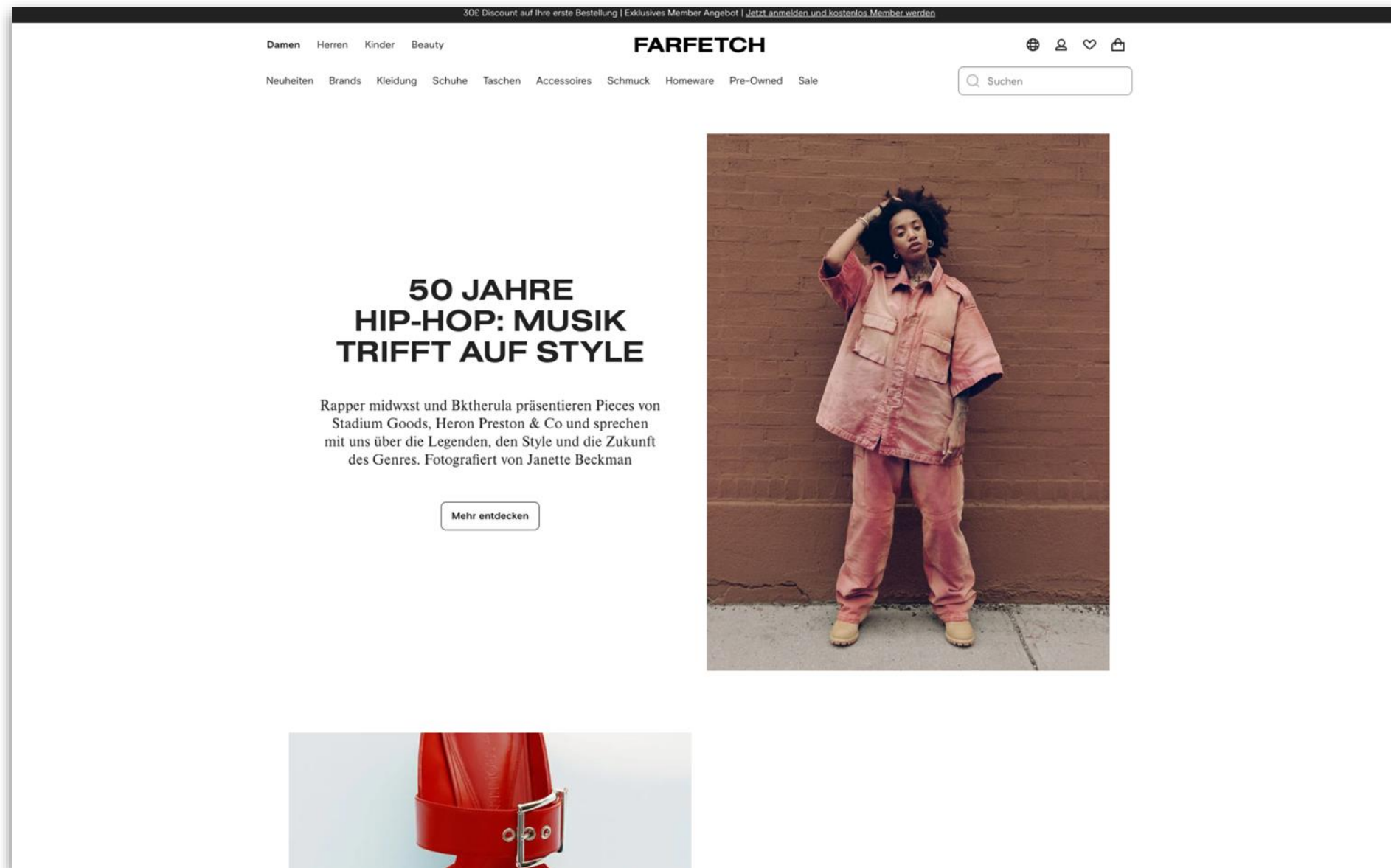
- Nike kann jetzt gezielte Marketingkampagnen und Produktveröffentlichungen schneller und effizienter durchführen
- Einfachere Integration von Technologie Partnern
- Durch die Umsetzung der Headless Strategie konnte Nike seinen Marktanteil erhöhen
- Headless Ansatz war damit ein großer Erfolg

10

Headless, API-First & Co.
Beispiel API-First

Headless und API-First

Beispiel API-First



- Einzigartigkeit von Farfetch: Netzwerk an Boutiquen, die zentralisiert Produkte anbieten
- Dafür braucht Farfetch eine robuste und flexible technologische Infrastruktur

Headless und API-First

Beispiel API-First - Umsetzung

- Farfetch stellt sicher, dass alle Funktionen über APIs zugänglich sind
- Damit könne Boutiquen ihre Systeme einfach mit Farfetch integrieren
- Der API-First Ansatz erleichtert auch Partnerschaften
- Z.B. kann Farfetch problemlos mit Luxusmarken oder anderen Technologie-Anbietern zusammenarbeiten

Headless und API-First

Beispiel API-First - Ergebnisse

- Schnelle Expansion und Integration neuer Boutiquen und Marken auf der Plattform
- Dadurch kamen auch stärkere und produktivere Partnerschaften zustande
- Durch die Nutzung von APIs kann Farfetch seine Angebote personalisieren und den Kunden ein einzigartiges Erlebnis bieten
- Echtzeit Aktualisierungen vom Inventar der Boutiquen

11

Headless, API-First & Co.
Bewerten & entscheiden

Headless und API-First

Bewerten & entscheiden

01

Unternehmensziele & Vision

- | Klare Definition langfristiger Ziele und Visionen
- | Flexibilität und zukunftssichere Systeme als Priorität?
- | Entscheidungsfindung basierend auf strategischer Ausrichtung



Headless und API-First

Bewerten & entscheiden



02

Technologische Reife und Ressourcen

- | Bewertung des aktuellen technologischen Standes
 - | Sind technologieorientierte Ressourcen und Fachwissen vorhanden?
 - | Bereitschaft zur Bewältigung potenzieller Implementierungsherausforderungen bewerten
- 

Headless und API-First

Bewerten & entscheiden



03

Anforderungen der Kundenerfahrung



- | Individualisiertes Kundenerlebnis als Ziel?
- | Bedarf an Multi-Plattform-Erlebnissen?
- | Notwendigkeit Touchpoints und API-Integrationen?

Headless und API-First

Bewerten & entscheiden

◀ 04 ▶

Wachstumspläne und Skalierung

- | Zukunftspläne für Geschäftserweiterung und Markteintritt
- | Suche nach leicht erweiterbaren Architekturen
- | Vermeidung von Überholungen oder Neugestaltungen

Headless und API-First Entscheidung treffen

◀ 05

Budgets, Kosten und Nutzen

- | Anfängliche Investitionen gegen langfristige Vorteile abwägen
- | Bewertung potenzieller Initialkosten bei Umstellung
- | Langfristiger Wert für das Unternehmen?

12

Headless, API-First & Co.
Aufgaben