Analyse und Planung eines Custom E-Commerce Shopsystems

Technologie-Evaluierung, Feature-Analyse
und Architekturempfehlung

B2C & B2B – Weltweiter Einsatz

Dokumenttyp:	Technische Analyse
Version:	1.0
Datum:	April 2026
Status:	Entwurf

Dieses Dokument dient als Entscheidungsgrundlage für die Technologieauswahl und Feature-Priorisierung eines maßgeschneiderten E-Commerce-Systems.

Inhaltsverzeichnis

1 Einleitung

1.1 Projektzielsetzung

Ziel dieses Projekts ist die Entwicklung eines maßgeschneiderten E-Commerce-Shopsystems, das sowohl B2C- (Business-to-Consumer) als auch B2B-Geschäftsmodelle (Business-to-Business) unterstützt. Das System soll weltweit einsetzbar sein und sich durch eine moderne, skalierbare Architektur auszeichnen.

Im Gegensatz zu bestehenden SaaS-Lösungen (Shopify, WooCommerce) oder Open-Source-Plattformen (Magento, PrestaShop) bietet ein Custom-System folgende Vorteile:

- **Volle Kontrolle** über Funktionalität, Datenhoheit und Geschäftslogik
- **Keine GMV-Gebühren** oder plattformabhängige Einschränkungen
- **Maßgeschneiderte B2B-Workflows**, die in Standard-Plattformen oft fehlen
- **Skalierbarkeit** ohne Plattform-Limits
- **Unabhängigkeit** von Drittanbieter-Roadmaps und Lizenzänderungen

1.2 Systemarchitektur – Überblick

Das System besteht aus drei Hauptkomponenten:

Backend RESTful API (optional GraphQL) mit Geschäftslogik, Datenbankzugriff, Authentifizierung und Integration externer Dienste (Zahlungen, Versand, Steuern).

Frontend Server-Side-Rendered (SSR) Webshop für Endkunden (B2C) und Geschäftskunden (B2B) mit optimierter Performance und SEO.

Admin Verwaltungsoberfläche für Produkte, Bestellungen, Kunden, Einstellungen und Reporting – separates Deployment mit eigenem Berechtigungssystem.

1.3 Methodik

Die Technologiebewertung in diesem Dokument basiert auf folgenden Kriterien:

1. **Ecosystem-Größe** – Verfügbarkeit von Paketen, Plugins und Community-Support
2. **TypeScript-Unterstützung** – Typsicherheit für weniger Fehler und bessere DX
3. **Skalierbarkeit** – Fähigkeit, mit wachsender Last umzugehen
4. **Time-to-Market** – Geschwindigkeit der Entwicklung bis zum MVP
5. **B2B-Eignung** – Unterstützung komplexer Geschäftslogik
6. **Internationalisierung** – Multi-Sprache, Multi-Währung, Multi-Region
7. **Entwickler-Verfügbarkeit** – Einstellbarkeit von qualifizierten Entwicklern
8. **Langfristige Wartbarkeit** – Clean Code, Testbarkeit, Modularität

Features werden nach dem MoSCoW-Prinzip priorisiert: **PFLICHT** (Pflicht), **SOLL** (Soll), **KANN** (Kann) und **NICHT BENÖTIGT** (Nicht benötigt).

2 Technologie-Analyse

2.1 Backend-Frameworks

Die Wahl des Backend-Frameworks bestimmt maßgeblich die Architektur, Entwicklungsgeschwindigkeit und Skalierbarkeit des gesamten Systems. Für ein E-Commerce-System mit hohen Anforderungen an Datenintegrität, API-Performance und B2B-Geschäftslogik wurden fünf Frameworks evaluiert.

Tabelle 1: Backend-Framework-Vergleich

Kriterium	**NestJS**	**FastAPI**	**Django**	**Go (Gin)**	**Rust (Actix)**
Sprache	TypeScript	Python	Python	Go	Rust
Typsicherheit	Hoch	Mittel	Gering	Hoch	Sehr hoch
Performance	Gut	Gut	Mittel	Sehr gut	Exzellent
Ecosystem	Sehr groß	Groß	Sehr groß	Mittel	Klein
Lernkurve	Mittel	Niedrig	Mittel	Mittel	Hoch
B2B-Eignung	Sehr gut	Gut	Gut	Gut	Gut
Time-to-Market	Schnell	Schnell	Mittel	Mittel	Langsam
E-Commerce-Ref.	Medusa, Vendure	Saleor	Saleor	–	–

NestJS (Node.js/TypeScript) ist die ausgereifteste Wahl für ein modernes E-Commerce-Backend. Es bietet Enterprise-Patterns wie Dependency Injection, modulare Architektur, Guards und Interceptors. Die TypeScript-Durchgängigkeit mit dem Frontend eliminiert Kontextwechsel. Medusa.js und Vendure – zwei führende Open-Source-E-Commerce-Plattformen – basieren auf diesem Stack.

FastAPI (Python) überzeugt durch schnelle Entwicklung und native Async-Unterstützung. Ideal, wenn KI/ML-Features (Produktempfehlungen, Preisoptimierung) im Vordergrund stehen.

Django (Python) bietet ein „Batteries-included"-Konzept mit eingebautem ORM, Admin-Panel und Sicherheitsfeatures. Saleor, eine der reifsten Open-Source-Commerce-Plattformen, nutzt Django als Basis.

Go und **Rust** bieten überlegene Rohperformance, eignen sich jedoch eher für spezialisierte Microservices (Zahlungsverarbeitung, Echtzeit-Inventar) als für die gesamte Anwendungsschicht. Die kleineren Ecosystems erhöhen den Entwicklungsaufwand.

2.2 Frontend-Frameworks

Für ein weltweit eingesetztes E-Commerce-Frontend sind Server-Side Rendering (SSR), SEO-Optimierung und Performance entscheidend. Vier Frameworks wurden verglichen.

Next.js (React) ist der klare Favorit für E-Commerce:

- **Incremental Static Regeneration (ISR)** ermöglicht gecachte Produktseiten mit automatischer Aktualisierung
- **Server Components** reduzieren die Bundle-Größe und verbessern die Ladezeit

Tabelle 2: Frontend-Framework-Vergleich

Kriterium	**Next.js (React)**	**Nuxt (Vue)**	**SvelteKit**	**Angular**
SSR/SSG	Exzellent (ISR)	Sehr gut	Gut	Gut
Ecosystem	Sehr groß	Groß	Wachsend	Groß
TypeScript	Nativ	Nativ	Nativ	Nativ
Bundle-Größe	Mittel	Mittel	Sehr klein	Groß
SEO	Exzellent	Sehr gut	Gut	Mittel
Komponentenbib.	Sehr viele	Viele	Wenige	Viele
Lernkurve	Mittel	Niedrig	Niedrig	Hoch
Arbeitsmarkt	Sehr groß	Groß	Klein	Groß
E-Commerce-Eig.	Exzellent	Sehr gut	Gut	Mittel

- **Image Optimization** ist eingebaut – kritisch für Produktfotografie
- Der **größte Arbeitsmarkt** vereinfacht die Teambildung

Nuxt (Vue) ist eine starke Alternative mit einfacherer Lernkurve, besonders in Europa beliebt.

SvelteKit liefert die kleinsten Bundle-Größen – ideal für Mobile-First-Strategien – hat aber ein kleineres Ecosystem und weniger E-Commerce-spezifische Bibliotheken.

2.3 Datenbanken

E-Commerce-Daten erfordern ACID-Konformität (Zahlungen, Bestellungen), flexible Attributspeicherung (Produktvarianten) und performante Suchabfragen.

Tabelle 3: Datenbank-Vergleich

Kriterium	**PostgreSQL**	**MySQL 8.x**	**MongoDB**
ACID-Konformität	Vollständig	Vollständig	Eingeschränkt
JSON-Unterstützung	JSONB (indexiert)	JSON (eingeschränkt)	Nativ
Volltextsuche	Eingebaut	Eingebaut	Eingebaut
Relationale Integrität	Exzellent	Gut	Keine (FK)
Skalierbarkeit	Vertikal + Read-Replicas	Vertikal + Replikation	Horizontal (Sharding)
Analytische Queries	Sehr performant	Mittel	Langsam
Multi-Tenancy	Schemas möglich	Datenbanken	Collections
E-Commerce-Eignung	Exzellent	Gut	Eingeschränkt
ORM-Support	Prisma, TypeORM	Prisma, TypeORM	Mongoose

PostgreSQL ist die empfohlene Wahl:

- JSONB-Spalten ermöglichen flexible Produktattribute bei voller Indexierung
- ACID-Konformität sichert finanzielle Datenintegrität
- 65–80% kürzere Ausführungszeiten bei komplexen Filterabfragen gegenüber MongoDB

- Alle führenden E-Commerce-Plattformen (Medusa, Saleor, Vendure) nutzen PostgreSQL

MongoDB eignet sich *nicht* als primäre Datenbank für E-Commerce aufgrund fehlender referentieller Integrität und langsamerer analytischer Abfragen. Die Flexibilität für Produktvarianten wird durch PostgreSQL JSONB gleichwertig abgedeckt.

2.4 Suchmaschinen

Eine leistungsfähige Produktsuche ist umsatzkritisch. Kunden, die die Suche nutzen, konvertieren 2–3x häufiger als Browser. Vier Suchmaschinen wurden verglichen.

Tabelle 4: Such-Engine-Vergleich

Kriterium	**Meilisearch**	**Elasticsearch**	**Algolia**	**Typesense**
Antwortzeit	<50ms	>100ms	<100ms	<100ms
Setup	Einfach	Komplex	SaaS (einfach)	Einfach
Self-Hosted	Ja	Ja	Nein (SaaS)	Ja
Kosten	Kostenlos (OSS)	Kostenlos (OSS)	Teuer	Günstig
Tippfehler-Tol.	Ja (Standard)	Konfigurierbar	Ja	Ja
Facettierung	Ja	Ja (mächtig)	Ja	Ja
Mehrsprachigkeit	Gut	Sehr gut	Sehr gut	Gut
Eignung	Ideal für E-Com.	Enterprise	Premium	Gute Alternative

Meilisearch ist die Empfehlung für den Start: In Rust geschrieben, liefert es Antwortzeiten unter 50ms mit eingebauter Tippfehlertoleranz. Self-Hosted und kostenlos, mit einfacher REST-API. Für Enterprise-Szenarien mit 10M+ Produkten kann später auf Elasticsearch gewechselt werden.

2.5 Zahlungsanbieter

Für weltweiten Einsatz muss das System mehrere Zahlungsanbieter unterstützen, da regionale Präferenzen stark variieren (SEPA in EU, Alipay in Asien, ACH in USA).

Tabelle 5: Payment-Provider-Vergleich

Kriterium	**Stripe**	**Mollie**	**Adyen**	**PayPal**
Länder	195+	EU-fokussiert	187	Weltweit
Zahlungsmethoden	Sehr viele	EU-Methoden	250+	PayPal + Karten
B2B (Rechnung)	Stripe Billing	Klarna, SEPA	Ja	Nein
API-Qualität	Exzellent	Sehr gut	Gut	Mittel
PCI-Compliance	SAQ-A	SAQ-A	SAQ-A	SAQ-A
Gebühren (EU)	1,5% + 0,25	1,2–2,9%	Interchange++	1,5–3,5%
Marktplatz-Support	Connect	–	for Platforms	–
Eignung	Primär	EU-Ergänzung	Enterprise	Ergänzung

Empfehlung: Stripe als primärer Provider (beste API, weltweite Abdeckung), Mollie als Ergänzung für europäische Zahlungsmethoden (iDEAL, Bancontact, EPS, Giropay). Für B2B-Kauf auf Rechnung: Stripe Billing oder Klarna B2B über Mollie.

Wichtig: Keine eigene Zahlungsabwicklung implementieren – die PCI-DSS-Anforderungen sind zu aufwändig. Stattdessen Tokenisierung über den jeweiligen Provider nutzen (SAQ-A Level).

2.6 Infrastruktur & Deployment

Tabelle 6: Infrastruktur-Vergleich

Kriterium	Hetzner	DigitalOcean	AWS	GCP
Kosten	Sehr günstig	Günstig	Mittel–Hoch	Mittel
EU-Rechenzentren	Ja (DE, FI)	Ja	Ja	Ja
DSGVO	Exzellent (DE)	Gut	Gut	Gut
Managed Services	Begrenzt	Mittel	Sehr viele	Viele
Skalierbarkeit	Manuell	Gut	Exzellent	Exzellent
Kubernetes	Nein	Ja (DOKS)	Ja (EKS)	Ja (GKE)
Eignung Start	MVP/Start	Wachstum	Enterprise	Enterprise

Empfohlener Deployment-Stack:

- **Containerisierung:** Docker + Docker Compose (Standard für alle Umgebungen)
- **CI/CD:** GitHub Actions (automatisierte Tests, Linting, Deployment)
- **Phase 1:** Hetzner Cloud oder DigitalOcean (kosteneffizient, EU-Rechenzentren)
- **Phase 2+:** Migration zu AWS ECS oder Kubernetes bei Bedarf
- **CDN:** Cloudflare (kostenloser Tier, DDoS-Schutz, globale Edge-Server)
- **Dateispeicher:** S3-kompatibel (MinIO self-hosted oder AWS S3)

3 Empfohlener Technologie-Stack

3.1 Gewichtete Entscheidungsmatrix

Vier Stack-Kombinationen wurden nach neun Kriterien bewertet (Skala 1–10). Die Gewichtung reflektiert die Prioritäten des Projekts: schnelle Markteinführung, B2B-Fähigkeit und weltweite Einsetzbarkeit.

Tabelle 7: Gewichtete Entscheidungsmatrix – Technologie-Stacks

Kriterium	**Gew.**	**Stack 1: NestJS + Next.js + PG**	**Stack 2: FastAPI + React + PG**	**Stack 3: Nuxt + Node + Mongo**	**Stack 4: Go + React + PG**
Time-to-Market	20%	9 (1,80)	7 (1,40)	8 (1,60)	5 (1,00)
TypeScript-Durchgäng.	15%	10 (1,50)	5 (0,75)	8 (1,20)	4 (0,60)
Ecosystem & Pakete	15%	9 (1,35)	8 (1,20)	7 (1,05)	6 (0,90)
Perform. & Skalierb.	15%	7 (1,05)	7 (1,05)	6 (0,90)	10 (1,50)
B2B-Eignung	10%	9 (0,90)	8 (0,80)	6 (0,60)	7 (0,70)
Entwickler-Verfügb.	10%	9 (0,90)	8 (0,80)	7 (0,70)	5 (0,50)
AI/ML-Integration	5%	6 (0,30)	10 (0,50)	5 (0,25)	6 (0,30)
Community & Support	5%	9 (0,45)	8 (0,40)	7 (0,35)	6 (0,30)
Wartbarkeit	5%	8 (0,40)	7 (0,35)	7 (0,35)	8 (0,40)
Gesamtpunktzahl	**100%**	**8,65**	**7,25**	**7,00**	**6,20**

3.2 Empfehlung: Stack 1

NestJS + Next.js + PostgreSQL + Redis + Meilisearch + Stripe/Mollie

Begründung:

1. **Eine Sprache (TypeScript)** über den gesamten Stack – Backend, Frontend, Admin. Kein Kontextwechsel, gemeinsame Typdefinitionen, Code-Sharing zwischen Schichten.
2. **NestJS** liefert Enterprise-Architektur (Dependency Injection, Module, Guards, Interceptors) – ideal für komplexe B2B-Geschäftslogik wie Genehmigungsworkflows und kundenspezifische Preisgestaltung.
3. **Next.js** bietet SSR, ISR und Server Components für optimale SEO und Performance – kritisch für Produktseiten und internationale Reichweite.
4. **Größtes Ecosystem** und Arbeitsmarkt – einfachere Teambildung und schnellere Problemlösung durch Community-Support.
5. **Direkte Referenzarchitekturen**: Medusa.js (Node.js/TypeScript) und Vendure (NestJS/TypeScript) als bewährte Vorbilder.

3.3 Detaillierter Stack-Überblick

Tabelle 8: Detaillierter Technologie-Stack

Schicht	Technologie	Version	Bemerkung
Backend	NestJS + TypeScript	10.x+	Modulare Architektur, DI
Frontend (Shop)	Next.js + React	15.x+	App Router, Server Components
Frontend (Admin)	Next.js + React	15.x+	Separates Deployment
ORM	Prisma	6.x+	Type-safe Database Access
Datenbank	PostgreSQL	16+	JSONB für flexible Attribute
Cache	Redis	7+	Sessions, Cart, Rate-Limiting
Suche	Meilisearch	1.x+	Produkt- und Inhaltssuche
Zahlungen	Stripe + Mollie	–	Stripe primär, Mollie für EU
Job-Queue	BullMQ	5.x+	E-Mails, Importe, Exporte
Dateispeicher	S3-kompatibel	–	MinIO oder AWS S3
Deployment	Docker Compose	–	Kubernetes optional später
CI/CD	GitHub Actions	–	Automatisierte Tests + Deploy
CDN	Cloudflare	–	Edge-Caching, DDoS-Schutz
Validierung	Zod	3.x+	TypeScript-native Validierung
Testing	Jest + Playwright	–	Unit-, Integration-, E2E-Tests

4 Feature-Analyse

Dieses Kapitel listet alle Features nach Priorität geordnet. Jede Kategorie enthält konkrete Einzelfeatures mit Zuordnung zu den Systemkomponenten (Backend, Frontend, Admin).

4.1 Pflicht-Features (PFLICHT)

Die folgenden 13 Feature-Kategorien sind für ein funktionsfähiges E-Commerce-System unabdingbar. Ohne sie kann kein Verkauf stattfinden oder das System entspricht nicht den gesetzlichen Anforderungen.

Tabelle 9: Pflicht-Features – Übersicht

Nr.	Kategorie	Komponente	Komplexität
P01	Produktverwaltung	Backend + Admin	Hoch
P02	Benutzer & Authentifizierung	Backend + Frontend	Mittel
P03	Warenkorb & Checkout	Alle	Hoch
P04	Bestellverwaltung	Backend + Admin	Hoch
P05	Zahlungsabwicklung	Backend + Frontend	Hoch
P06	Versand & Lieferung	Backend + Admin	Mittel
P07	Steuerberechnung	Backend	Hoch
P08	Sicherheit & DSGVO	Alle	Hoch
P09	Produktsuche	Backend + Frontend	Mittel
P10	Responsives Design	Frontend	Mittel
P11	Mehrsprachigkeit (i18n)	Alle	Mittel
P12	Mehrwährungsfähigkeit	Backend + Frontend	Mittel
P13	Admin-Panel	Admin	Hoch

4.1.1 P01 – Produktverwaltung

Grundlage jedes E-Commerce-Systems. Ohne Produkte kein Verkauf.

- Produktanlage mit Titel, Beschreibung (Rich Text), SKU, EAN/GTIN
- Varianten-Management (Größe, Farbe, Material) mit eigenen SKUs und Preisen
- Digitale und physische Produkte unterscheiden
- Produktbilder mit Galerie, Zoom und Alt-Texten
- Hierarchische Kategorien und Tags
- Produktattribute (filterbar): Gewicht, Maße, Material, Marke
- Lagerbestandsverwaltung: Bestand pro Variante, Reservierung bei Checkout
- Produkt-Status: Entwurf, Aktiv, Archiviert
- Massenimport/-export (CSV/JSON)
- B2B: Kundenspezifische Produktsichtbarkeit (Katalogbeschränkung)

4.1.2 P02 – Benutzer & Authentifizierung

Erforderlich für Bestellhistorie, gespeicherte Adressen und Personalisierung.

- E-Mail/Passwort-Registrierung mit E-Mail-Verifizierung
- Passwort-Zurücksetzen (tokenbasiert, zeitlich begrenzt)
- Optionaler Social Login (Google, Apple)
- Kundenprofil: Name, Adressen (Rechnung/Lieferung), Telefon
- B2B: Firmenkonten mit Mitarbeiterverwaltung und Rollenkonzept
- Gastbestellung (ohne Registrierung)
- Session-Management und sichere Token-Verwaltung (JWT + Refresh-Tokens)
- Rate-Limiting auf Login-Versuche (Brute-Force-Schutz)
- Multi-Faktor-Authentifizierung (MFA) für Admin-Zugänge

4.1.3 P03 – Warenkorb & Checkout

Der kritischste Conversion-Pfad. Jeder Reibungspunkt kostet Umsatz.

- Warenkorb: Hinzufügen, Entfernen, Mengenänderung
- Persistenter Warenkorb (Session + Datenbank für eingeloggte Nutzer)
- Warenkorb-Zusammenführung bei Login (Gast-Cart → User-Cart)
- Mehrstufiger oder Single-Page-Checkout
- Rechnungs-/Lieferadresse (getrennt oder identisch)
- Versandmethoden-Auswahl mit Echtzeitpreisen
- Zahlungsmethoden-Auswahl
- Gutscheincode-Eingabe
- Bestellübersicht vor Abschluss
- B2B: Bestellnummer (PO-Number) eingeben, Genehmigungsworkflow
- Warenkorbabbruch-Tracking für spätere Wiederansprache

4.1.4 P04 – Bestellverwaltung

Kerngeschäftsdatensatz. Rechtlich erforderliche Aufbewahrung der Bestellhistorie.

- Automatische Bestellnummer-Generierung
- Status-Workflow: Ausstehend → Bestätigt → In Bearbeitung → Versendet → Zugestellt
- Kundenseitige Statusanzeige mit Sendungsverfolgung
- Admin: Bestellliste mit Filter, Suche, Sortierung
- Admin: Bestelldetails, manuelle Statusänderung

- Nachbestellung aus Bestellhistorie (Re-Order)
- B2B: Erweiterte Status (Genehmigung ausstehend, Freigegeben, Gesperrt)
- Rechnungserstellung (PDF) pro Bestellung
- Bestellnotizen (intern und kundensichtbar)

4.1.5 P05 – Zahlungsabwicklung

Unterstützung mehrerer Zahlungsmethoden ist für weltweiten Einsatz unerlässlich.

- Stripe-Integration: Kreditkarte, SEPA, Apple Pay, Google Pay
- Mollie-Integration: iDEAL, Bancontact, EPS, Giropay, Klarna
- PayPal als optionaler dritter Provider
- Provider-Abstraktion: einheitliches Interface für alle Zahlungsanbieter
- Tokenisierung (keine Kartendaten im System speichern)
- 3D Secure 2.0 für Kreditkartenzahlungen
- Webhook-Verarbeitung für asynchrone Zahlungsbestätigungen
- Teil- und Vollrückerstattungen
- B2B: Kauf auf Rechnung (Net 30/60/90), SEPA-Lastschrift
- Mehrwährungsfähige Zahlungen

4.1.6 P06 – Versand & Lieferung

Verschiedene Produkte, Gewichte und Zielorte erfordern unterschiedliche Versandoptionen.

- Mehrere Versanddienstleister (DHL, DPD, UPS, FedEx, lokale Carrier)
- Echtzeit-Versandkostenberechnung oder feste Versandkostentabellen
- Adressvalidierung (länderspezifisch)
- Versandetikett-Generierung
- Sendungsverfolgungsnummern mit Carrier-Tracking-Links
- Kostenloser Versand ab Bestellwert (konfigurierbar)
- B2B: Speditionsversand für Großbestellungen
- Internationale Versandregeln und Zolldeklarationen

4.1.7 P07 – Steuerberechnung

Gesetzliche Pflicht. Fehlerhafte Steuerberechnung führt zu Bußgeldern.

- Automatische MwSt-Berechnung nach Kundenstandort
- EU-MwSt: 15–27% je nach Land, OSS-Regelung für digitale Güter
- US Sales Tax: Bundesstaat- und City-Level (2000+ Jurisdiktionen)

- UK VAT: 20% Standard nach Brexit
- Steuersätze pro Produktkategorie (Normalsatz, ermäßigt, befreit)
- B2B: Reverse-Charge-Verfahren bei gültiger USt-IdNr
- USt-IdNr-Validierung über VIES (EU)
- Steuerberichte für Buchhaltung (Export CSV/PDF)
- Integration mit TaxJar oder Avalara für automatisierte Steuerberechnung

4.1.8 P08 – Sicherheit & DSGVO

Nicht verhandelbar. DSGVO-Verstöße: bis zu 20 Mio. EUR oder 4% des Jahresumsatzes.

- HTTPS/TLS 1.3 für gesamte Kommunikation
- Passwort-Hashing mit bcrypt oder Argon2
- Verschlüsselung sensibler Daten (PII) in der Datenbank
- DSGVO: Einwilligungsmanagement, Cookie-Banner, Datenschutzerklärung
- Recht auf Löschung (Konto- und Datenlöschung auf Anfrage)
- Datenportabilität (Export persönlicher Daten als JSON/CSV)
- Audit-Logging aller Admin-Aktionen
- CORS, CSP, HSTS und weitere Security-Header
- Rate-Limiting und DDoS-Schutz
- Regelmäßige Dependency-Updates (Sicherheitspatches)

4.1.9 P09 – Produktsuche

Kunden erwarten schnelle, relevante Suchergebnisse.

- Volltextsuche über Produktnamen, Beschreibungen, SKUs
- Autocomplete/Suchvorschläge während der Eingabe
- Tippfehlertoleranz (Fuzzy-Matching)
- Antwortzeit unter 200ms
- Ergebnis-Paginierung
- B2B: Suche nach SKU, Artikelnummer, interner Bezeichnung
- Indexierung über Meilisearch mit automatischer Synchronisation

4.1.10 P10 – Responsives Design

Über 50% des E-Commerce-Traffics kommt von Mobilgeräten.

- Mobile-First responsive Design
- Touch-optimierte Buttons und Interaktionen

- Lazy Loading für Bilder
- Optimierte Ladezeit auf 3G/4G-Verbindungen (<3 Sekunden)
- Mobile-Wallet-Support (Apple Pay, Google Pay)
- Testen auf verschiedenen Geräten und Bildschirmgrößen

4.1.11 P11 – Mehrsprachigkeit (i18n)

60% der Onlinekäufer kaufen selten auf englischsprachigen Seiten. 13% höhere Conversion bei muttersprachlicher Darstellung.

- UI-Lokalisierung: alle Texte, Fehlermeldungen, E-Mails
- Produktbeschreibungen in mehreren Sprachen
- Spracherkennung (Accept-Language Header + GeoIP)
- Sprachumschalter ohne Session-Verlust
- URL-Struktur: `/de/produkte`, `/en/products`, `/fr/produits`
- RTL-Unterstützung für Arabisch
- i18next oder react-intl als Frontend-Bibliothek
- Anbindung an Übersetzungsmanagement-Plattform (Crowdin, Lokalise)

4.1.12 P12 – Mehrwährungsfähigkeit

Kunden erwarten Preise in ihrer lokalen Währung.

- Automatische Währungserkennung (GeoIP-basiert)
- Manueller Währungsumschalter
- Basiswährung + automatische Umrechnung über Echtzeit-Wechselkurse
- Alternativ: feste Preise pro Währung (wichtig für B2B-Verträge)
- Korrekte Währungsformatierung pro Locale ($100.00 vs. 100,00 vs. ¥100)
- Unterstützung gängiger Währungen: EUR, USD, GBP, CHF, JPY, CNY etc.
- Währungsspezifische Zahlungsabwicklung

4.1.13 P13 – Admin-Panel

Zentrale Verwaltungsoberfläche für den täglichen Betrieb.

- Dashboard: Umsatz, Bestellungen, Neukunden, Topseller
- Produktverwaltung: CRUD, Massenbearbeitung, Bildupload
- Bestellverwaltung: Liste, Detail, Statusänderung, Rückerstattung
- Kundenverwaltung: Übersicht, Bestellhistorie, Notizen
- Einstellungen: Shop-Name, Währung, Steuersätze, Versandoptionen

- E-Mail-Template-Verwaltung
- Benutzerrollen und Berechtigungen (Admin, Manager, Support)
- Einfache Berichte (Umsatz nach Zeitraum, Kategorie, Kunde)
- Export-Funktionen (CSV/PDF)

4.2 Soll-Features (SOLL)

Diese 12 Feature-Kategorien verbessern das System erheblich und sind für ein wettbewerbsfähiges Produkt wichtig. Sie sollten in Phase 2 implementiert werden.

Tabelle 10: Soll-Features – Übersicht

Nr.	Kategorie	Komponente	Komplexität
S01	B2B-Preisgestaltung	Backend + Admin	Hoch
S02	Erweiterte Suche & Filter	Backend + Frontend	Mittel
S03	Aktionen & Rabatte	Backend + Admin	Mittel
S04	Wunschliste	Backend + Frontend	Niedrig
S05	Bewertungen & Rezensionen	Backend + Frontend	Mittel
S06	E-Mail-Benachrichtigungen	Backend	Mittel
S07	SEO-Optimierung	Frontend + Backend	Mittel
S08	Analytics & Reporting	Admin + Backend	Mittel
S09	CMS / Blog	Alle	Mittel
S10	Multi-Warehouse-Inventar	Backend + Admin	Hoch
S11	Retouren & RMA	Backend + Admin	Mittel
S12	REST/GraphQL API	Backend	Mittel

4.2.1 S01 – B2B-Preisgestaltung

Kern-Feature für B2B. Ohne kundenspezifische Preise kein ernsthafter B2B-Betrieb.

- Kundengruppen-basierte Preise (Händler, Großhändler, VIP)
- Kundenspezifische Preislisten (vertraglich vereinbart)
- Staffelpreise: 1–10 Stk. → 10 , 11–50 Stk. → 9 , 50+ Stk. → 8
- Rabatte auf Kategorie- oder Produktebene pro Kundengruppe
- Netto-Preisanzeige für B2B-Kunden, Brutto für B2C
- Mindestbestellmenge (MOQ) pro Produkt konfigurierbar
- Angebotsanfrage (RFQ) für individuelle Preisverhandlung

4.2.2 S02 – Erweiterte Suche & Filter

Verbessert die Conversion um 10–20% durch gezieltes Finden.

- Facettierte Suche: Preis, Kategorie, Marke, Farbe, Größe, Bewertung
- Dynamische Filteranzahl (zeigt verfügbare Produkte pro Filter)

- Sortierung: Relevanz, Preis, Neuheit, Beliebtheit, Bewertung
- Gespeicherte Suchen (besonders für B2B-Wiederholungskäufe)
- Filterbare Produktlisten in Kategorieseiten

4.2.3 S03 – Aktionen & Rabatte

Marketing-Werkzeug zur Umsatzsteigerung und Kundengewinnung.

- Gutscheincodes: prozentual, absolut, kostenloser Versand
- Einmal- oder Mehrfachverwendung, ablaufdatumbasiert
- Automatische Kampagnen: Zeitraum, Kategorie, Mindestbestellwert
- Warenkorb-Aktionen: „Kaufe 2, erhalte 3. gratis“
- Bundle-Rabatte: Produktkombinationen günstiger
- Treuestufen: Rabatt basierend auf Gesamtumsatz des Kunden

4.2.4 S04 – Wunschliste

Erhöht Wiederbesuchsrate. Kunden speichern Produkte für späteren Kauf.

- Produkte zur Wunschliste hinzufügen/entfernen
- Wunschliste teilen (URL-basiert)
- Benachrichtigung bei Preissenkung oder Wiederverfügbarkeit
- Produktvergleich (2–5 Produkte nebeneinander)

4.2.5 S05 – Bewertungen & Rezensionen

Social Proof steigert Conversion. Bewertungen sind wertvoller SEO-Content.

- 5-Sterne-Bewertung + Freitextrezension
- Nur nach Kaufabschluss bewertbar (verifizierte Käufe)
- Admin-Moderation vor Veröffentlichung
- „Hilfreich“-Markierung durch andere Kunden
- Durchschnittsbewertung auf Produktseite und in Suchergebnissen
- Antwortfunktion für den Shopbetreiber

4.2.6 S06 – E-Mail-Benachrichtigungen

Reduziert Support-Anfragen und hält Kunden informiert.

- Bestellbestätigung (sofort nach Bestellabschluss)
- Zahlungsbestätigung
- Versandbestätigung mit Tracking-Link

- Zustellungsbenachrichtigung
- Passwort-Zurücksetzen, Kontobestätigung
- Optionale Warenkorbabbruch-E-Mail
- Template-Engine mit Variablen (Bestellnummer, Kundenname, Produkte)
- Anbindung an E-Mail-Service (SendGrid, AWS SES, Brevo)

4.2.7 S07 – SEO-Optimierung

Kritisch für organisches Wachstum. Produktseiten müssen in Suchmaschinen gefunden werden.

- Individuelle Meta-Tags (Title, Description) pro Seite
- Saubere URL-Struktur: `/produkte/rote-laufschuhe`
- Automatische XML-Sitemap-Generierung
- Structured Data / JSON-LD (Product, BreadcrumbList, Organization)
- Canonical-Tags zur Vermeidung von Duplicate Content
- Open Graph und Twitter Cards für Social Sharing
- Bildoptimierung mit Alt-Texten und WebP-Format
- hreflang-Tags für mehrsprachige Seiten

4.2.8 S08 – Analytics & Reporting

Datenbasierte Entscheidungen für Sortiment, Preise und Marketing.

- Umsatz-Dashboard: Tages-, Wochen-, Monatsumsatz, YoY-Vergleich
- Top-Produkte, Kategorie-Performance
- Kundenanalyse: Neukunden vs. Wiederkäufer, Customer Lifetime Value
- Bestandsberichte: Slow-Mover, Out-of-Stock-Rate
- Conversion-Funnel: Besucher → Warenkorb → Checkout → Kauf
- Google Analytics 4 (GA4) Integration
- Export in CSV/PDF für Buchhaltung und Management

4.2.9 S09 – CMS / Blog

Content-Marketing treibt organischen Traffic und baut Autorität auf.

- Statische Seiten (Über uns, Kontakt, AGB, Datenschutz)
- WYSIWYG-Editor für Seiteninhalt
- Blog mit Kategorien, Tags, Autor, Veröffentlichungsdatum
- Zeitgesteuerte Veröffentlichung (Scheduled Publishing)

- Wiederverwendbare Content-Blöcke (Testimonials, Feature-Boxen)
- SEO-Felder pro Seite/Beitrag

4.2.10 S10 – Multi-Warehouse-Inventar

Für skalierende Unternehmen mit mehreren Lagerstandorten.

- Mehrere Lagerorte anlegen
- Bestand pro Variante und Lagerort verwalten
- Automatische Lagerort-Auswahl bei Bestellung (Nähe zum Kunden)
- Lagerumzüge zwischen Standorten tracken
- Verfügbarkeitsanzeige: verfügbar, reserviert, verkaufbar
- Nachbestellpunkte und Bestandsalarme

4.2.11 S11 – Retouren & RMA

Kunden erwarten einen einfachen Rückgabeprozess. Gute Retourenpolitik erhöht Conversion.

- Retourenanfrage aus Bestellhistorie erstellen
- RMA-Nummern-Generierung
- Retourenetikett-Generierung
- Rückgabefrist konfigurierbar (z.B. 30 Tage)
- Teilretouren (einzelne Artikel einer Bestellung)
- Rückerstattung oder Gutschrift nach Wareneingang
- Zustandsbewertung zurückgesendeter Ware

4.2.12 S12 – REST/GraphQL API

Ermöglicht Mobile Apps, Drittanbieter-Integrationen und headless Betrieb.

- REST API für alle Kernressourcen (Produkte, Bestellungen, Kunden, Cart)
- OpenAPI/Swagger-Dokumentation (automatisch generiert)
- Optional: GraphQL-Endpunkt für flexiblere Frontend-Abfragen
- API-Versionierung (v1, v2) für Abwärtskompatibilität
- API-Key-Authentifizierung und OAuth 2.0
- Rate-Limiting pro API-Key
- Webhooks für Events (Bestellung erstellt, Zahlung bestätigt, Versand)
- Webhook-Retry-Mechanismus und Logging

4.3 Kann-Features (KANN)

Diese 11 Feature-Kategorien sind Nice-to-have und sollten in Phase 3 oder bedarfsgesteuert implementiert werden. Sie bieten Mehrwert, sind aber nicht geschäftskritisch.

Tabelle 11: Kann-Features – Übersicht

Nr.	Kategorie	Komponente	Komplexität
K01	Treueprogramm / Bonuspunkte	Alle	Mittel
K02	Abonnements / Wiederkehrende Best.	Backend + Frontend	Hoch
K03	Marktplatz (Multi-Vendor)	Alle	Sehr hoch
K04	KI-gestützte Empfehlungen	Backend + Frontend	Hoch
K05	Live-Chat / Chatbot	Frontend	Mittel
K06	Social-Media-Integration	Frontend	Niedrig
K07	Geschenkkarten	Backend + Frontend	Mittel
K08	EDI / PunchOut (Enterprise)	Backend	Hoch
K09	A/B-Testing	Frontend + Backend	Mittel
K10	PWA-Support	Frontend	Mittel
K11	Multi-Tenant / Multi-Store	Alle	Sehr hoch

4.3.1 K01 – Treueprogramm / Bonuspunkte

Steigert Customer Lifetime Value und Wiederkaufrate.

- Punktesystem: 1 Punkt pro 1 Umsatz
- Punkte einlösen gegen Rabatte oder Gratisprodukte
- Ablaufdatum für Punkte (konfigurierbar)
- Mitgliedsstufen: Bronze, Silber, Gold mit steigenden Vorteilen
- Empfehlungsprogramm: Belohnung für Weiterempfehlung

4.3.2 K02 – Abonnements / Wiederkehrende Bestellungen

Wiederkehrender Umsatz, Kundenkomfort bei Verbrauchsartikeln.

- Abo-Produkte mit flexiblen Intervallen (wöchentlich, monatlich, quartalsweise)
- Pausieren, Überspringen, Kündigen durch Kunden
- Automatische Verlängerung und Zahlung
- Erinnerung vor Verlängerung
- Mengenänderung zwischen Zyklen

4.3.3 K03 – Marktplatz (Multi-Vendor)

Erweitert das Sortiment ohne eigenes Inventar. Erfordert erheblichen Zusatzaufwand.

- Vendor-Registrierung und -Freischaltung
- Vendor-Dashboard: eigene Produkte, Bestellungen, Umsatz

- Provisionsmodell (konfigurierbar pro Vendor)
- Admin-Genehmigung neuer Produkte
- Vendor-Auszahlungsmanagement
- Separate Vendor-Bewertungen

4.3.4 K04 – KI-gestützte Empfehlungen

Erhöht den durchschnittlichen Bestellwert durch personalisierte Vorschläge.

- „Ähnliche Produkte“ auf Produktseiten
- „Häufig zusammen gekauft“ im Warenkorb
- Personalisierte Empfehlungen basierend auf Kauf- und Browsinghistorie
- Collaborative Filtering oder Content-Based Algorithmen
- Empfehlungen in E-Mails

4.3.5 K05 – Live-Chat / Chatbot

Echtzeit-Support verbessert Kundenzufriedenheit und reduziert Warenkorbabbrüche.

- Chat-Widget auf der Website
- KI-Chatbot für häufige Fragen (FAQ, Bestellstatus)
- Eskalation an menschlichen Mitarbeiter
- Offline-Nachrichtenerfassung
- Chat-Verlaufshistorie

4.3.6 K06 – Social-Media-Integration

Kostenloser Marketing-Kanal durch Social Sharing.

- Social Login (Facebook, Google, Apple)
- Produkt-Teilen auf Social Media
- Instagram/TikTok Shopping-Integration
- Social-Media-Pixel für Remarketing (Meta, TikTok)

4.3.7 K07 – Geschenkkarten

Zusätzliche Umsatzquelle und Geschenkoption.

- Digitale Geschenkkarten mit festen oder individuellen Beträgen
- E-Mail-Zustellung mit personalisierter Nachricht
- Einlösung im Checkout
- Restguthaben-Verwaltung

- Optionales Ablaufdatum

4.3.8 K08 – EDI / PunchOut (Enterprise B2B)

Für Integration mit Großkunden-Beschaffungssystemen (SAP Ariba, Coupa).

- EDI-Nachrichtenformate (X12, EDIFACT) empfangen und verarbeiten
- PunchOut-Katalog: Produktkatalog über cXML bereitstellen
- Automatische Bestellverarbeitung aus Procurement-Systemen
- Authentifizierung gegenüber Einkaufssystem des Kunden

4.3.9 K09 – A/B-Testing

Datengesteuerte Optimierung von Conversion-Raten.

- Test-Framework: Varianten von UI, Preisen, Content ausspielen
- Traffic-Splitting zwischen Varianten
- Statistische Auswertung (Signifikanzniveau)
- Gewinner-Deklaration und automatisches Rollout

4.3.10 K10 – PWA-Support

App-ähnliche Erfahrung ohne App-Store-Installation.

- Service Worker für Offline-Funktionalität
- Install-Prompt (Manifest.json, App-Icon, Splashscreen)
- Push-Notifications für Bestellstatus und Angebote
- Cache-Strategien für häufig besuchte Seiten

4.3.11 K11 – Multi-Tenant / Multi-Store

Mehrere Shops/Marken aus einer Instanz betreiben. Hohe architektonische Komplexität.

- Separate Stores mit eigenen Produkten, Preisen, Einstellungen
- Datenisolierung pro Store (Kunden, Bestellungen)
- Geteilte Ressourcen: Produktkatalog mit store-spezifischen Preisen
- Ein Admin-Panel für alle Stores
- Store-spezifische Domains, Währungen, Sprachen

4.4 Nicht benötigte Features (NICHT BENÖTIGT)

Die folgenden Funktionen liegen außerhalb des Systemscopes. Sie werden besser durch spezialisierte Drittanbieter abgedeckt und per API angebunden.

Tabelle 12: Abgrenzung – Nicht benötigte Features

Kategorie	**Begründung**	**Alternative (Integration)**
Eigene Buchhaltung / ERP	Zu komplex, regulatorisch anspruchsvoll, schnell veraltet. Besser als Integration.	DATEV, sevDesk, lexoffice, Xero, NetSuite
Vollständiges WMS	Lagerverwaltung (Picking, Packing, Routenoptimierung) ist hochspezialisiert.	ShipHero, Fulfillmenttools, externe WMS per API
Eigene Zahlungsabwicklung	PCI-DSS Level 1 erfordert jährliche Audits ($50K–$500K). Unverhältnismäßig.	Stripe, Mollie, Adyen (Tokenisierung)
E-Mail-Marketing-Plattform	Spezialisierte Tools bieten Segmentierung, Automation und Analytics auf Enterprise-Niveau.	Mailchimp, Brevo (ex-Sendinblue), Klaviyo
Social-Media-Management	Nicht Kernaufgabe eines Shopsystems. Separater Workflow und Tools.	Buffer, Hootsuite, Later
Vollständiges CRM	Kundenbeziehungsmanagement umfasst Sales-Pipeline, Ticketing, Anrufmanagement.	HubSpot, Salesforce, Pipedrive

Prinzip: Das Shopsystem stellt für jede dieser Kategorien Schnittstellen (REST API, Webhooks) bereit, über die Drittanbieter angebunden werden. Es implementiert diese Funktionen nicht selbst.

5 B2B-spezifische Anforderungen

Der gleichzeitige Support von B2C und B2B ist eine der größten Herausforderungen des Systems. B2B-Kunden haben fundamental andere Anforderungen als Endverbraucher.

5.1 B2B vs. B2C – Fundamentale Unterschiede

Tabelle 13: B2B vs. B2C – Feature-Vergleich

Aspekt	B2C	B2B	Implementierung
Preisgestaltung	Einheitlich für alle Kunden	Kundenspezifisch, vertraglich	Preisgruppen-Engine, Preislisten
Preisanzeige	Brutto (inkl. MwSt)	Netto (zzgl. MwSt)	Toggle basierend auf Kontotyp
Bestellprozess	Sofortkauf, keine Genehmigung	Genehmigungspflichtig ab Betrag X	Workflow-Engine mit Eskalation
Zahlung	Sofort (Karte, PayPal)	Auf Rechnung (Net 30/60/90)	Kreditlimit-Prüfung, Invoicing
Konto	Einzelperson	Firma mit mehreren Mitarbeitern	Company-Account-Hierarchie
Bestellmenge	Einzelstücke	Großmengen, Paletten	MOQ, Staffelpreise, Schnellbestellung
Katalog	Gesamtes Sortiment	Kundenspezifischer Katalog	Katalogzuweisung pro Kundengruppe
Kaufentscheidung	Emotional, schnell	Rational, mehrere Entscheider	Angebotssystem, Verhandlung

5.2 Firmenkonten & Rollenmanagement

- **Company Account:** Übergeordnetes Konto mit Firmendaten (Name, USt-IdNr, Adresse, Zahlungsbedingungen)
- **Mitarbeiter-Accounts:** Zugeordnet zum Company Account, mit individuellen Rollen:

  Einkäufer Kann bestellen bis zu einem Betragslimit

  Genehmiger Kann Bestellungen freigeben

  Admin Kann Mitarbeiter verwalten, Rollen zuweisen

  Finanz Kann Rechnungen einsehen und Zahlungen verwalten

  Viewer Nur Lesezugriff auf Bestellhistorie und Katalog
- **Einladungsworkflow:** Admin lädt Mitarbeiter per E-Mail ein
- **Budgetlimits:** Pro Mitarbeiter oder Abteilung konfigurierbar

5.3 Genehmigungsworkflows

1. Einkäufer legt Bestellung an und übermittelt zur Genehmigung

2. System prüft: Bestellwert > Limit → Genehmigung erforderlich
3. Genehmiger erhält E-Mail-Benachrichtigung
4. Genehmiger: Freigeben, Ablehnen oder Kommentieren
5. Bei Mehrstufigkeit: Eskalation an nächste Ebene
6. Nach Freigabe: automatische Zahlungsauslösung und Bestellverarbeitung

5.4 Zahlungsbedingungen

- **Kauf auf Rechnung:** Net 30, Net 60, Net 90 (pro Kunde konfigurierbar)
- **Kreditlimit:** Maximaler offener Rechnungsbetrag pro Firma
- **SEPA-Lastschrift:** Einmaliges Mandat, wiederkehrende Einzüge
- **Vorkasse:** Überweisung mit automatischer Zuordnung (IBAN-Referenz)
- **Skonto:** 2% Rabatt bei Zahlung innerhalb von 10 Tagen
- **Zahlungserinnerungen:** Automatisiert bei Fälligkeit und Überfälligkeit

5.5 Schnellbestellung & CSV-Upload

B2B-Kunden kennen ihre Artikelnummern und wollen schnell große Bestellungen aufgeben:

- **Quick-Order-Formular:** SKU + Menge eingeben, direkt in den Warenkorb
- **CSV-Upload:** Bestellliste hochladen (SKU, Menge) für Massenbestellungen
- **Bestellvorlagen:** Wiederkehrende Bestellungen als Template speichern
- **Nachbestellung:** Ein-Klick-Wiederholung vergangener Bestellungen

6 Internationalisierung

Für weltweiten Einsatz muss das System von Beginn an auf Internationalisierung ausgelegt sein. Nachträgliches Hinzufügen ist um ein Vielfaches aufwändiger.

6.1 Übersicht der Anforderungen

Tabelle 14: Internationalisierungsanforderungen

Aspekt	Ansatz	Technologie
Übersetzungen	Key-Value mit Pluralisierung und ICU Message Format	i18next / next-intl
Währungen	Echtzeit-Kurse + manuell gepflegte Preise pro Währung	Open Exchange Rates API
Datumsformate	Locale-basierte Formatierung	Intl API (ECMAScript)
Zahlenformate	Locale-basiert (1.000,50 vs. 1,000.50)	Intl.NumberFormat
Adressen	Länderspezifische Adressfelder und Validierung	Google Address Validation API
Rechtstexte	Separate AGB, Widerrufsbelehrung, Datenschutz pro Land	CMS mit Sprachzuordnung
Maßeinheiten	Metrisch (EU) vs. Imperial (US, UK)	Konfiguration pro Region
RTL-Support	Arabisch, Hebräisch: Right-to-Left Layout	CSS logical properties, dir-Attribut

6.2 Mehrsprachigkeit (i18n)

URL-Strategie: Sprachpräfix im Pfad (empfohlen für SEO):

```
example.com/de/produkte/rote-schuhe
example.com/en/products/red-shoes
example.com/fr/produits/chaussures-rouges
```

Vorteile: Eine Domain, zentralisierte Domain-Autorität, einfachere Verwaltung als Subdomains.

Spracherkennung (Priorität):

1. URL-Pfad (`/de/...`)
2. Gespeicherte Benutzerpräferenz (Cookie/Account)
3. Accept-Language Header des Browsers
4. GeoIP-basierter Fallback

Übersetzungsmanagement:

- Entwickler pflegen Schlüssel in JSON-Dateien (Standardsprache: Englisch)
- Professionelle Übersetzer arbeiten über Plattform (Crowdin, Lokalise)
- Automatischer Pull der Übersetzungen ins Repository via CI/CD
- Produktbeschreibungen: mehrsprachige Felder in der Datenbank

6.3 Mehrwährungsfähigkeit

Zwei Strategien (beide unterstützen):

Automatische Umrechnung Basiswährung (EUR) + Echtzeit-Wechselkurse. Gut für B2C mit vielen Märkten. Preise können zwischen Seitenbesuchen schwanken.

Feste Preise pro Währung Manuell gepflegte Preise in jeder Währung. Wichtig für B2B-Verträge und Preiskonsistenz. Höherer Pflegeaufwand.

Währungsformatierung muss locale-korrekt sein:

- USA: $1,000.00
- Deutschland: 1.000,00
- Japan: ¥1,000 (keine Dezimalstellen)
- Schweiz: CHF 1'000.00

6.4 Regionale Rechtsanforderungen

- **EU:** DSGVO, Widerrufsbelehrung, Preisangabenverordnung (Grundpreis), Impressumspflicht, Cookie-Consent
- **Deutschland:** Buttonlösung („Zahlungspflichtig bestellen"), Fernabsatzgesetz, Verpackungsgesetz
- **UK:** UK GDPR (post-Brexit), Consumer Rights Act, VAT-Regeln
- **USA:** CCPA/CPRA (Kalifornien), CAN-SPAM Act, keine einheitliche Datenschutzregelung
- **Weitere:** LGPD (Brasilien), PIPA (Südkorea), APPI (Japan)

7 Sicherheit & Compliance

7.1 DSGVO / GDPR

Die DSGVO gilt für alle Unternehmen, die personenbezogene Daten von EU-Bürgern verarbeiten – unabhängig vom Firmensitz.

Kernprinzipien und ihre Umsetzung im System:

- **Datenminimierung:** Nur erforderliche Daten erheben. Keine Pflichtfelder für nicht-benötigte Informationen.
- **Zweckbindung:** Daten nur für den angegebenen Zweck verwenden. Klare Trennung von Bestelldaten und Marketingdaten.
- **Einwilligungsmanagement:** Explizites Opt-in für Newsletter, Cookie-Tracking, personalisierte Werbung. Kein Pre-Checked-Checkbox.
- **Recht auf Löschung:** Kundenkonto vollständig löschbar. Bestelldaten anonymisieren (gesetzliche Aufbewahrungsfristen beachten).
- **Datenportabilität:** Export aller persönlichen Daten als JSON/CSV.
- **Auftragsverarbeitung:** DPA (Data Processing Agreement) mit allen Drittanbietern (Stripe, Meilisearch Cloud, etc.).
- **Datenschutzerklärung:** Detailliert, verständlich, pro Sprache/Land.

7.2 PCI-DSS Compliance

Da keine Kartendaten im System gespeichert oder verarbeitet werden (Tokenisierung über Stripe/Mollie), gilt **SAQ-A** – das niedrigste Compliance-Level:

- Zahlungsformulare werden als iFrames der Payment-Provider eingebettet
- Keine Kartendaten berühren das eigene Backend
- TLS 1.2+ für alle Verbindungen
- Regelmäßige Vulnerability-Scans

7.3 OWASP Top 10 – Maßnahmen

7.4 Authentifizierung & Autorisierung

JWT + Refresh-Token Access-Token (15 Min. Gültigkeit) + Refresh-Token (7 Tage, in httpOnly-Cookie). Token-Rotation bei jedem Refresh.

OAuth 2.0 Für API-Zugriff durch Drittanbieter. Authorization Code Flow mit PKCE für öffentliche Clients.

RBAC Role-Based Access Control mit feingranularen Berechtigungen. Rollen: Super-Admin, Shop-Admin, Support, B2B-Admin, B2B-Einkäufer, B2B-Genehmiger.

MFA Pflicht für Admin-Zugänge. Optional (empfohlen) für B2B-Genehmiger. TOTP-basiert (Google Authenticator, Authy).

Tabelle 15: Sicherheitsmaßnahmen gegen OWASP Top 10

Bedrohung	**Maßnahme**	**Implementierung**
Injection	Parameterisierte Queries, kein String-Concatenation	Prisma ORM (prepared statements)
Broken Auth	Sichere Token-Verwaltung, MFA für Admin	JWT + Refresh-Token, Passport.js
XSS	Output-Encoding, Content Security Policy	React (auto-escaping), Helmet.js
CSRF	SameSite-Cookies, CSRF-Tokens	NestJS CSRF-Guard
SSRF	URL-Validierung, Allowlist für externe Requests	Input-Validierung mit Zod
Security Misc.	Security-Header, Dependency-Updates	Helmet.js, Dependabot/Renovate
Insecure Design	Threat Modeling, Security Reviews	Architektur-Review vor Deployment
Broken Access	RBAC, Ressourcen-basierte Autorisierung	NestJS Guards + CASL-Bibliothek
Crypto Failures	Starke Hashing-Algorithmen, Verschlüsselung	Argon2 (Passwörter), AES-256 (PII)
Logging/Monitor.	Zentrale Logs, Alerting bei Anomalien	Winston + Loki oder CloudWatch

8 Open-Source-Referenzarchitekturen

Drei führende Open-Source-E-Commerce-Plattformen dienen als Referenz für Architekturentscheidungen. Ziel ist nicht die Übernahme einer Plattform, sondern das Lernen aus bewährten Mustern.

8.1 Learnings für unser System

Von Medusa.js lernen: Modulares Plugin-System, bei dem jedes Modul (Produkte, Bestellungen, Zahlungen) unabhängig ersetzt werden kann. Multi-Region-Konzept für länderspezifische Einstellungen (Währung, Steuern, Lager).

Von Saleor lernen: GraphQL-Schema-Design für E-Commerce. Ausgereifte Multi-Currency/Multi-Language-Implementierung. Channel-basiertes Multi-Store-Konzept.

Von Vendure lernen: NestJS-spezifische Patterns: Dependency Injection für Service-Layer, Custom Decorators für Autorisierung, Event-Bus für lose Kopplung. State-Machine für Bestellstatus.

Wichtig: Keine dieser Plattformen als Fork übernehmen. Die Abweichung von der Upstream-Codebasis macht Updates nach 3–6 Monaten praktisch unmöglich. Stattdessen: eigenes System bauen, Patterns und API-Design als Referenz nutzen.

Tabelle 16: Open-Source-Referenzplattformen im Vergleich

Kriterium	**Medusa.js**	**Saleor**	**Vendure**
Sprache	TypeScript (Node.js)	Python (Django)	TypeScript (NestJS)
API	REST (+ GraphQL geplant)	GraphQL	GraphQL
Datenbank	PostgreSQL	PostgreSQL	PostgreSQL / MySQL
Architektur	Modular (Plugin-System)	Monolith (modular)	Plugin-basiert
Multi-Currency	Ja (Multi-Region)	Ja (sehr ausgereift)	Ja (Plugin)
B2B-Support	Begrenzt (Custom)	Begrenzt (Extension)	Plugin möglich
Admin-Panel	Eigenes (React)	Dashboard (React)	Angular Admin UI
GitHub-Stars	~31K	~21K	~6K
Stärken	Erweiterbarkeit, DX, größte Community	Multi-Channel, GraphQL, i18n out-of-the-box	Saubere NestJS-Architektur, Type-Safety
Schwächen	Weniger Features als Saleor, B2B braucht Custom-Code	Python (anderer Stack), kleinere Community	Kleinste Community, Angular-Admin
Relevanz	**Hoch** – gleicher Stack	Mittel – Architektur-Referenz	**Hoch** – NestJS-Referenz

9 Architektur-Übersicht

9.1 Architekturansatz: Modularer Monolith

Das System startet als **modularer Monolith** – ein einzelnes Deployment mit klar getrennten internen Modulen. Dieser Ansatz kombiniert:

- **Einfachheit eines Monolithen:** Ein Deployment, ein Repository, einfaches Debugging, keine Netzwerk-Latenz zwischen Modulen
- **Modularität für spätere Zerlegung:** Jedes Modul hat klare Grenzen (eigenes Interface, eigene DTOs), sodass es bei Bedarf als Microservice extrahiert werden kann

Kein Microservice-Start! Microservices verursachen bei kleinen Teams mehr Overhead als Nutzen (Network Calls, Distributed Tracing, Service Discovery, Deployment-Komplexität). Die Zerlegung erfolgt erst bei konkretem Bedarf.

9.2 System-Komponenten

9.3 API-Design

Primär: REST API mit klarer Ressourcen-Struktur:

```
GET    /api/v1/products          # Produktliste (paginiert,
   filterbar)
GET    /api/v1/products/:id      # Produktdetails
POST   /api/v1/cart/items        # Artikel zum Warenkorb
   hinzufuegen
POST   /api/v1/checkout          # Checkout starten
```

Tabelle 17: Architektur-Komponenten

Komponente	Technologie	Verantwortlichkeit
API-Server	NestJS	REST-Endpunkte, Middleware, Guards, Validierung
Produkt-Modul	NestJS Module	Katalog, Varianten, Kategorien, Attribute, Suche
Bestell-Modul	NestJS Module	Cart, Checkout, Bestellungen, Status-Workflow
Zahlungs-Modul	NestJS Module	Provider-Abstraktion (Strategy Pattern), Webhooks
Kunden-Modul	NestJS Module	Auth, Profile, Firmenkonten, Rollen
B2B-Modul	NestJS Module	Preislisten, Genehmigungen, Zahlungsbedingungen
Versand-Modul	NestJS Module	Carrier-Integration, Tracking, Etiketten
Such-Service	Meilisearch	Produkt-Indexierung und -Abfragen
Cache-Layer	Redis	Sessions, Warenkörbe, Rate-Limiting, Caching
Job-Queue	BullMQ (Redis)	E-Mails, Import/Export, Bildverarbeitung
Datei-Speicher	S3 / MinIO	Produktbilder, Dokumente, Exporte
Event-Bus	NestJS EventEmitter	Lose Kopplung zwischen Modulen

```
GET    /api/v1/orders              # Bestellhistorie
POST   /api/v1/orders/:id/return # Retoure anfragen
```

Optional: GraphQL als zweiter Endpunkt für Frontend-Entwickler, die flexiblere Abfragen benötigen (z.B. Produktseite mit Reviews, Related Products und Inventory in einem Request).

Webhooks für asynchrone Events:

```
POST /webhooks/stripe         # Stripe-Zahlungsbestaetigung
POST /webhooks/shipping       # Carrier-Tracking-Updates
```

9.4 Datenbank-Schema (Kernentitäten)

```
Produkt          1 --- N  Variante
Produkt          N --- M  Kategorie
Variante         1 --- N  Lagereintrag (pro Lagerort)
Kunde            1 --- N  Adresse
Kunde            1 --- N  Bestellung
Kunde            N --- 1  Firma (B2B)
Firma            1 --- N  Mitarbeiter (Kunde mit Rolle)
Bestellung       1 --- N  Bestellposition
Bestellung       1 --- 1  Zahlung
Bestellung       1 --- N  Versand
Bestellung       1 --- N  Rechnung
Preisliste       N --- M  Kundengruppe
Preisliste       1 --- N  Preislisteneintrag (Produkt + Preis)
```

9.5 Event-Driven Architecture

Module kommunizieren über Events statt direkter Aufrufe:

```
order.created     -> E-Mail-Service (Bestellbestaetigung senden)
```

```
order.created      -> Inventar-Service (Bestand reservieren)
payment.confirmed  -> Bestell-Service (Status aktualisieren)
order.shipped      -> E-Mail-Service (Versandbestaetigung senden)
order.shipped      -> Tracking-Service (Tracking-Updates starten)
```

Vorteil: Module kennen einander nicht direkt. Neue Features (z.B. Analytics, Loyalty-Punkte) können als Listener hinzugefügt werden, ohne bestehenden Code zu ändern.

9.6 Caching-Strategie

Produktkatalog Redis-Cache mit TTL (5–15 Min.), Invalidierung bei Admin-Änderung

Warenkorb Redis mit Session-ID als Key. TTL: 7 Tage (Gast), unbegrenzt (eingeloggt)

Suchergebnisse Meilisearch hat eigenes Caching. Zusätzlich CDN-Caching für häufige Queries

Seitencaching Next.js ISR für Produktseiten, Kategorieseiten, Blog-Beiträge

API-Responses ETag-basiertes Caching für unveränderliche Ressourcen

10 Zusammenfassung & Empfehlung

10.1 Empfohlener Stack (Zusammenfassung)

Backend: NestJS + TypeScript + Prisma + PostgreSQL
Frontend: Next.js + React + TypeScript
Admin: Next.js + React (separates Deployment)
Suche: Meilisearch **Cache:** Redis **Queue:** BullMQ
Zahlungen: Stripe (primär) + Mollie (EU)
Deployment: Docker Compose + GitHub Actions + Cloudflare CDN

10.2 Phasenplan

Tabelle 18: Implementierungs-Phasenplan

Phase	Zeitraum	Features	Team
Phase 1 (MVP)	Monat 1–6	Alle Pflicht-Features (P01–P13): Produktverwaltung, Auth, Cart/Checkout, Bestellungen, Zahlungen, Versand, Steuern, Sicherheit, Suche, Responsive Design, i18n, Multi-Currency, Admin-Panel	3–4 Entwickler
Phase 2	Monat 7–12	Priorisierte Soll-Features: B2B-Preisgestaltung (S01), Erweiterte Suche (S02), Rabatte (S03), E-Mail-Benachrichtigungen (S06), SEO (S07), Analytics (S08), API (S12)	3–5 Entwickler
Phase 3	Ab Monat 13	Verbleibende Soll-Features (S04, S05, S09–S11) und bedarfsgesteuerte Kann-Features (K01–K11)	4–6 Entwickler

10.3 Nächste Schritte

Nach Genehmigung dieses Analysedokuments folgen:

1. **Detaillierte technische Spezifikation** – Datenbankschema, API-Endpunkte, Modulstruktur
2. **UI/UX-Design** – Wireframes und Prototypen für Shop, Checkout, Admin
3. **Projektsetup** – Repository, CI/CD-Pipeline, Docker-Konfiguration, Coding-Standards, Linting
4. **Sprint-Planung** – Aufgaben für Phase 1 in 2-Wochen-Sprints aufteilen
5. **Infrastruktur** – Hosting einrichten, Datenbank provisionieren, Staging-Umgebung deployen