

# BlackDress - System-Architektur

**Version:** 1.2 **Erstellt:** 2026-05-08 **Letzte Aktualisierung:** 2026-05-08 abend (nach Build-Tag A-Z + Step 1/3-3/3) **Status:** Pre-Live (Site live mit Pre-Launch-Gate, Live-Schalter wartet auf 13.5.+) **Build-Tag-Schluss-Stand:** 72 HTML-Pages, Email-Worker-Skelett ready (provider-agnostic), Test-Suite 13/13 grün, Schema.org Phase 2 mit 160 JSON-LD-Types, Pages-Index für Schnellzugang gebaut

## 1 · Site-Map (72 HTML-Pages — Stand 2026-05-08 nach Step P)

### Hauptnavigation

Pfad	Zweck
<code>/index.html</code>	Homepage / Hero / Featured Products
<code>/shop.html</code>	Produkt-Liste mit Filter (Kategorie, Größe, Farbe)
<code>/product.html?id=&lt;sku&gt;</code>	Produkt-Detail (Gallery, Variants, Material, Stock, Add-to-Cart)
<code>/sale.html</code>	Sale-Items
<code>/lookbook.html</code>	Visual-Inspiration-Page
<code>/kategorien.html</code>	Kategorie-Übersicht
<code>/about.html</code>	Brand-Story

### Account / Konto

Pfad	Zweck
<code>/login.html</code>	Login-Form (Phase F: Step E)
<code>/register.html</code>	Account-Anlage (optional, Guest-Checkout möglich)
<code>/password-reset.html</code>	Password-Reset (Phase F)
<code>/konto.html</code>	Konto-Übersicht (Bestellungen, Adresse)
<code>/wishlist.html</code>	Wishlist (LocalStorage + opt. server-side wenn logged in)

### Checkout / Bestellung

Pfad	Zweck
<code>/checkout.html</code>	Checkout-Flow (Step F — Step-0 Login/Guest + Mock-Payrexx-Modal + bd_last_order persist)

<code>/bestellung-bestaetigt.html</code> ✓ NEU(Step L)	Order-Confirmation (Hero+Next-Steps+Items+Address+Trust+CTA, lädt aus bd_last_order)
<code>/bestellung-status.html</code>	Order-Status-Page (Token-basiert)
<code>/newsletter-bestaetigt.html</code>	Newsletter-Opt-In-Confirmation

## Information / Guides

Pfad	Zweck
<code>/buying-guides.html</code>	Übersicht aller Kauf-Guides
<code>/guide-bh-groesse.html</code>	BH-Größen-Anleitung mit Tabelle
<code>/guide-erste-bestellung.html</code>	First-Order-Guide
<code>/guide-korsett.html</code>	Korsett-Tragen-Guide
<code>/guide-material.html</code>	Material-Übersicht
<code>/guide-stil.html</code>	Stil-Beratung
<code>/groesstabelle.html</code>	Übergreifende Größentabelle
<code>/pflege.html</code>	Pflege-Hinweise
<code>/versand.html</code>	Versand-Infos
<code>/faq.html</code>	FAQ

## Trust / Brand

Pfad	Zweck
<code>/charity.html</code>	Krebsliga-Spende-Info
<code>/loyalty.html</code>	Treueprogramm
<code>/blog.html</code> + <code>/blog/*</code>	Brand-Blog
<code>/lookbook.html</code>	Visual-Lookbook
<code>/partners.html</code>	Partner / Vendor-Liste
<code>/mystery.html</code>	Mystery-Box-Konzept
<code>/geschenkkarte.html</code>	Geschenkkarten-Verkauf

## Legal (Skeleton aktiv — Step G — warten auf Anwaltsdokumente 13.5.)

Pfad	Zweck
------	-------

<code>/agb.html</code>	AGB-Skeleton (Anwaltstexte ausstehend)
<code>/datenschutz.html</code>	Datenschutz-Skeleton (DSGVO/nDSG)
<code>/impressum.html</code>	Impressum-Skeleton
<code>/widerruf.html</code>	Widerrufsrecht (CH OR konform)
<code>/cookie-einstellungen.html</code> ✓ NEU(Step J)	Cookie-Settings — DSGVO/nDSG-konform, jederzeit anpassbar via window.BD_Cookie API
<code>/zu-jung.html</code>	Altersverifikations-Hinweis (18+)

## Error-Pages (Step O)

Pfad	Zweck
<code>/404.html</code> ✓ NEU	404 mit cream BG, Search-Autofocus, 4 Quicklinks
<code>/500.html</code> ✓ NEU	Server-Error mit Retry-Pattern
<code>/error.html</code> ✓ NEU	Generic-Error mit ?msg= URL-Param-Support (XSS-safe escape)

## Kontakt / Sonstiges

Pfad	Zweck
<code>/kontakt.html</code>	Kontakt-Formular
<code>/404.html</code>	Fehlerseite

## Pre-Launch

- `/gate.js` (in jedem HTML eingebunden) — Pre-Launch-Password-Gate (PBKDF2-SHA256 250k iter, Pw `BlackDress-2026!`)

## Cookie / Privacy (Step J)

- `/cookie-banner.js` (in jedem HTML eingebunden) — DSGVO-konformes Cookie-Banner (4 Kategorien, Opt-In default für non-essentials)
- `window.BD_Cookie` JS-API: `.show()` / `.get()` / `.clear()` / `.has(category)`
- LocalStorage-Key: `bd_cookie_consent_v1`
- Settings-Page: `/cookie-einstellungen.html`

## 2 · File-Struktur auf VPS (Stand nach Step P)

```

/root/blackdress/
├── *.html (72 Seiten – siehe Site-Map)
├── gate.js + gate.js.src # Pre-Launch-Gate (minified) + Source
├── cookie-banner.js + cookie-banner.js.src # Cookie-Banner (Step J)

```

```

├─ sw.js + sw.js.src           # Service-Worker (Caching)
├─ manifest.webmanifest       # PWA-Manifest
├─ _headers                   # ✓NEU(Step P) Cloudflare Pages Cache-Headers
├─ robots.txt                 # noindex bis Live
├─ sitemap.xml                # SEO-Sitemap (58 URLs www.blackdress.ch +
lastmod)
├─ build_minify.py            # ✓NEU(Step P) CSS/JS-Minify-Pipeline
(idempotent)
|
├─ assets/
|   ├─ style.css + style.css.src # Hauptstyle (minified) + Source
|   └─ app.js + app.js.src      # Globale JS-Logik (Cart Schema v2 + 7d-expire)
+ Source
|   ├─ logo.png                # Master-Logo
|   ├─ og-blackdress.jpg      # ✓NEU OG-Image 1200×630 (alle 72 Pages)
|   └─ about-hero.jpg          # CSS-Background für about.html (NICHT OG-
Image!)
|   └─ product-images/        # 1687 Produkt-Foto-Verzeichnis (extern via CF
Pages)
|
├─ data/
|   ├─ products.json           # 1687 Produkte mit Variants + material (Step
audit-v2 + Variante C)
|   ├─ audit_report.json      # Material-Audit-Report
|   └─ audit_report.md        # Markdown-Version
|
├─ d1/
|   ├─ schema.sql             # Cloudflare D1 Schema (10 Tabellen)
|   └─ README.md              # Deploy-Anleitung – TEST-DB live (id
42aa3e86-...)
|
├─ lib/
|   ├─ aes.js + aes.js.src     # AES-256-GCM Encryption (Worker-kompatibel)
|   └─ aes.test.mjs           # Smoke-Tests (6/6 grün)
|
├─ email_templates/          # ✓Step E – 12 Files (3 templates × 4 langs
DE/FR/IT/EN)
|   ├─ order_confirmation.{de,fr,it,en}.html
|   ├─ shipping_notification.{de,fr,it,en}.html
|   ├─ cancellation.{de,fr,it,en}.html
|   └─ README.md
|
├─ test_plans/
|   ├─ e2e_test_plan.md        # 58 Cases / 8 Suiten
|   └─ playwright/            # ✓NEU(Step H) 8 specs + helpers +
playwright.config.ts (58 stubs)
|
├─ manual/                   # ← DIESE Dokumente
|   ├─ blackdress_zugangs_inventar.{md,pdf}
|   ├─ blackdress_architektur.{md,pdf} (this file)
|   └─ blackdress_operations_handbuch.{md,pdf}
|
├─ backups/
|   └─ products_pre_*.json     # Pre-Edit-Snapshots

```

```

|
|─ blog/ # Blog-Inhalte
|
|─ 404.html / 500.html / error.html # ✓NEU(Step 0) Error-Pages
|
|─ workers/ # ✓NEU(Step R) CF-Worker-Code
|   └─ email/ # blackdress-email Worker (provider-agnostic)
|       └─ src/index.js # Entry + Routing + Bearer-Auth
|           └─ src/lib/ # emailService, orderService, template,
idempotency, aes
|               └─ src/providers/ # stub.js, brevo.js, cf-email-workers.js
|                   └─ test/idempotency.test.mjs # 7/7 grün
|                       └─ wrangler.toml # D1-binding + secrets-config
|                           └─ README.md # Architektur + Deploy-Anleitung
|
|─ test_plans/playwright/specs/ # 9 specs (Step H+Z) – 8 STUBS, 9 ACTIVE
|   └─ 01-browse-cart.spec.ts # 8/8 grün (Don 2026-05-08)
|       └─ 09-security-basics.spec.ts # 5/5 grün
|
|─ extract_material.py + audit_products.py # Genesis-Scripts

```

### Build-Pipeline (Step P):

- Source-of-Truth: `*.src` files (style.css.src, app.js.src, ...)
- Bei Edit: `*.src` ändern → `python3 build_minify.py` → minified `*.{css, js}` regeneriert
- Sizes: 140 KB → 113 KB (-19%); plus Cloudflare gzip/brotli edge → effective ~75% weniger Wire-Bytes

## 3 · Datenfluss: Kunde-Klick bis Bestellung-gespeichert

### Phase 1 — Browse (Public, anonymous)

1. Besucher → `https://www.blackdress.ch` → Cloudflare Pages CDN serviert HTML
2. `gate.js` prüft LocalStorage: ist Pw schon eingegeben? Ja → durchlassen. Nein → Password-Form
3. Browser fetcht `/data/products.json` (1687 Produkte mit Variants)
4. JS rendert Produkt-Liste (`shop.html`) oder Detail (`product.html?id=xxx`)
5. Add-to-Cart → LocalStorage `blackdress_cart`

### Phase 2 — Checkout (anonymous oder logged-in)

1. Besucher klickt "Checkout" → `/checkout.html`
2. Wahl: Login ODER Guest-Checkout (Step E baut beide)
3. Form-Daten: Name + Email + Adresse + Wahl Zahlung
4. JS validiert Pflichtfelder + PLZ-Format
5. Bei Submit:
  - **Email** wird AES-256-GCM verschlüsselt (lib/aes.js, Worker-Side)
  - **customer\_hash** = SHA256(email + salt) berechnet
  - Adresse wird AES-GCM verschlüsselt
6. Worker schreibt **D1**:

- `customers`-Eintrag (encrypted PII)
- `orders`-Eintrag mit `status='pending'` + 10-Jahre-Retention
- `order_items`-Einträge pro Variant
- `inventory.qty_reserved += 1` (vor Payment)

### Phase 3 — Payment (Payrex)

1. Worker initiiert Payrex-Session via API
2. Browser redirected zu Payrex-Hosted-Page (oder iframe)
3. Kunde bezahlt mit TWINT/Card/PayPal/Klarna
4. Payrex → Webhook → Cloudflare-Worker
5. Worker checkt Webhook-Signature
6. Bei Success: `orders.status='paid', inventory.qty_on_hand -= ordered_qty, qty_reserved -= 1`
7. Email-Worker triggert: `order_confirmation.de.html` rendern + senden

### Phase 4 — Fulfillment (manuell durch Don)

1. Don sieht neue Order in CF-D1 (via Don's Order-Dashboard oder direkt SQL)
2. Don packt + verschickt
3. Don markiert in D1: `orders.status='shipped', tracking_number=xxx, shipped_at=now()`
4. Email-Worker triggert: `shipping_notification.de.html` mit Tracking-Link

### Phase 5 — Post-Delivery

1. (Optional ab Phase D) NPS-Email nach 7 Tagen
2. (Optional ab Phase D) T+14/T+30/T+60 Repeat-Trigger via customer-lifecycle Skill

## 4 · Datenbanken / Speicher

### Cloudflare D1: `blackdress-prod` (geplant, noch nicht deployed)

Tabelle	Inhalt	PII	Retention
<code>schema_meta</code>	Migration-Version	nein	forever
<code>customers</code>	Hash + AES-encrypted email/ name	ja (encrypted)	10y oder bis Löschantrag
<code>customer_tags</code>	Lifecycle-Tags (vip, repeat, dormant)	nein	linked
<code>orders</code>	Order-Header + AES-encrypted Adresse	ja (encrypted)	<b>10 Jahre legal-Pflicht (CH OR / EU GoBD)</b>
<code>order_items</code>	Line-Items mit SKU-Snapshot	nein	linked
<code>inventory</code>	Stock-State	nein	live

<code>cart_sessions</code>	Pre-Payment-Carts	quasi-PII (token)	7d
<code>email_log</code>	Audit + Idempotency	minimal	2y
<code>discount_codes</code>	Codes mit max 20% Cap	nein	bis <code>valid_until</code> + 90d

## Cloudflare KV (geplant)

- `KNOWLEDGE_QA` Namespace (existiert für knowledge-imperium, NICHT BlackDress)
- BlackDress kein KV aktuell — möglicherweise Phase F für Session-State

## Cloudflare R2 (Bilder, falls migriert)

- **Aktuell:** Images in `/root/blackdress/assets/product-images/` direkt auf CF Pages
- **Geplant:** Migration zu R2 wenn  $1687 \times 5$  Images = ~8000 Files Pages-Limits sprengt

## Genesis Brain (separates System, NICHT BlackDress-spezifisch)

- `/root/genesis/brain/brain.db` — SQLite mit Lessons + Decisions + Facts
- BlackDress-relevante Einträge taggen mit `blackdress`

## File-System-Storage

- `/root/blackdress/data/products.json` — Master-Katalog (Genesis-edited, manuell oder via `extract_material.py`)
- `/root/blackdress/backups/` — pre-edit Snapshots vor jeder größeren Modifikation

## 5 · Externe Services — was macht wer

Service	Verantwortet
Cloudflare Pages	Static-Hosting der 37 HTML + assets, CDN, SSL, DDoS-Schutz
Cloudflare DNS (über Wix-DNS Plan B)	Routing <code>www.blackdress.ch</code> → Pages
Cloudflare D1	Customer + Order + Inventory-DB (geplant)
Cloudflare Workers	API-Endpoints für Checkout, Payrexx-Webhook, Email-Trigger
Cloudflare R2	Image-CDN (geplant nach Migration)
Wix	Domain-Registrar (NICHT mehr DNS-Hosting; nur Registrar-Funktion)
Google Workspace	Mail-Hosting für <code>info@blackdress.ch</code>
Payrexx	Zahlungsabwicklung (TWINT/Card/PayPal/Klarna) inkl. Refund-Verwaltung

<b>Email-Provider</b> (CF Email-Workers oder Brevo, Phase noch offen)	Transaktional-Emails (Confirmation/Shipping/Cancellation)
<b>fal.ai</b>	Hero-Shots / Lifestyle-Bilder via genesis-image (Cost-Approval-pflichtig)
<b>Anthropic</b>	Genesis-Operations (Claude Max Subscription, all-included)

## 6 · Crons (auf VPS via Claude-Code-CronCreate)

**Wichtig:** Diese Crons laufen in Genesis-Session (`systemctl restart claude` reset sie). SSoT in `/root/genesis/cron_renewer/desired_crons.json`. Nach Server-Restart muss CronRenewer (oder manuell) recreaten.

### BlackDress-relevante Crons (geplant)

Aktuell **keine** BlackDress-spezifischen Crons aktiv. Geplant nach Live-Gang:

Cron	Zweck	Schedule	Status
<code>blackdress_inventory_sync</code>	Sync inventory mit Vendor-Sources	(TBD)	nicht aktiv
<code>blackdress_cart_abandon_cleanup</code>	qty_reserved freigeben nach 24h	hourly	nicht aktiv
<code>blackdress_low_stock_alert</code>	Telegram-Alert wenn $\geq 10$ SKUs low-stock	daily	nicht aktiv

### System-Crons (BlackDress-tangential)

- `8954a619` HR-Audit daily 17:15 (Trading + Bot-Imperium-spezifisch)
- `c1f0a56b` CronRenewer daily 04:11 (renewt alle Crons)
- `785a9c64` Brain-Backup daily 02:07 (sichert Genesis-Brain inkl. blackdress-tagged entries)

**Logs-Pfad** (ad-hoc): `/var/log/syslog` für systemd-Events, Genesis-Conversation-Logs in `/root/.claude/projects/-root/<session-id>/`.

## 7 · Backups

Was	Wohin	Frequenz
<code>data/products.json</code>	<code>/root/blackdress/backups/products_pre_*.json</code> (vor jeder größeren Modifikation)	bei Bedarf, manuell vor Edit
<code>dashboard/data.json</code> (Bot-Imperium)	<code>/root/dashboard/backups/</code>	bei Bedarf vor Sync
<code>genesis/brain/brain.db</code>	<code>/root/genesis/brain/backups/brain-YYYY-MM-DD.db</code>	daily 02:07 (Cron <code>785a9c64</code> ), 7d Rotation

Hostinger VPS-Snapshots	Hostinger-Cloud (extern)	per Hostinger-Panel, manuell oder geplant
D1-Database	(über CF Pages Snapshot ODER <code>wrangler dl export</code> )	TBD nach Deploy
Memory-Files	<code>/root/genesis/backups/pre-restart-*/memory/</code> (vor Server-Restart)	manuell pre-Restart

**Pflicht-Snapshot vor jeder Schema-Edit oder Bulk-Modifikation** (Hard-Rule per genesis-protocol).

## 8 · Pre-Launch-Gate

- **Aktiv** auf allen 37 HTML-Pages via `gate.js`
- **Mechanik:** PBKDF2-SHA256 250k iterations, Hash in `gate.js` hardcoded
- **Passwort:** `BlackDress-2026!` (Don's PW-Speicherort: `<KeePassXC>`)
- **Bypass:** nach erfolgreichem PW: LocalStorage-Token `bd_gate_v1` gesetzt
- **Entfernen:** Live-Gang erfordert Deletion oder Disable von `gate.js` (TODO `remove-gate` im Register)

## 9 · Sub-Bot-Architektur (BlackDress-Touch)

Sub-Bot	Was	BlackDress-Bezug
<code>genesis</code> (CEO)	Orchestrator	alle Decisions + Pre-Mortems
<code>blackdress</code>	Domain-Specialist	Brand-Voice, Customer-Support, Newsletter
<code>genesis-image</code>	fal.ai-Bilder	Hero-Shots, Saisonale-Kollektionen
<code>genesis-knowledge</code>	RSS + X-Wall	Trend-Detection für saisonale Picks (Phase D Cross-Bot-Pattern 1)
<code>hr-bot</code>	Daily-Audit	inkl. BlackDress-Aktivität-Tracking
<code>trading-bot</code>	Solana-Memecoin-Paper	NICHT BlackDress (parallel-empire)

## Genesis Brain (separates System) — BlackDress-Touchpoints

- **Brain-Pfad:** `/root/genesis/brain/brain.db` (SQLite, WAL, 9 Tabellen)
- **Tabellen:** `schema_meta` + `facts/+tags` + `decisions/+tags` + `lessons/+tags`
- **API:** `BrainWrite/BrainRead/BrainQuery` (7 Functions: `facts_by_bot`, `facts_by_tag`, `latest_fact`, `lessons_by_bot`, `decisions_by_topic`, `search_text`, `count_by_tag`)
- **BlackDress-relevante Einträge:** tagged mit `blackdress` (z.B. `v2-PDF-print-friendly-Lesson`, `OG-Image-Cache-Bust-Pattern`, `Schema-Konsistenz-Hard-Gate`)
- **Backup:** daily 02:07 Cron `785a9c64` → `/root/genesis/brain/backups/brain-YYYY-MM-DD.db` (7d Rotation)

## Skills (BlackDress-Touchpoints)

Skill	BlackDress-Bezug
opportunity-vetting	Decision-Gate für neue BlackDress-Features (Score 0-100, <60 reject / 60-79 vet / ≥80 GO)
revenue-discipline	ROI-Tagging + Opportunity-Cost + Cost-Approval-Latenz-Fallback (>10 CHF)
cross-bot-synergy	6 Patterns (Bot A quiet, kopiert nicht) — z.B. genesis-knowledge → BlackDress saisonale Picks
customer-lifecycle	BlackDress-CRM mit SHA256+Salt customer_hash, T+7/T+14/T+30/T+60 Repeat-Trigger

## 10 · Build-Log Steps A–Z + 1/3-3/3 (2026-05-08)

Step	Was	Outcome
A	Mobile-Fix (Material-Block flex-wrap + @media)	Mobile-Layout korrekt
B	Domain-Routing (www.blackdress.ch)	Live mit Custom-Domain
C	Operations-Manual + 3 PDFs (v1)	Don-Independence-Docs
D	D1 Test-Mode Deploy (blackdress-test)	9 Tabellen migriert, ID 42aa3e86-...
E	Email-Templates FR/IT/EN (12 Files)	3 Templates × 4 Sprachen
F	Checkout-Flow Skeleton (Mock-Payrex + Step-0 Login/Guest)	bd_last_order persist
G	Legal-Page-Skeletons (AGB/ Datenschutz/Impressum/Widerruf)	Anwaltstexte folgen 13.5.+
H	E2E Playwright-Stubs (8 Specs / 58 Cases)	Test-Suite-Skeleton ready
I	SEO-Vorbereitung (sitemap.xml, robots.txt, OG-Tags bulk-inject)	58 URLs in sitemap
J	Cookie-Banner (4 Kategorien DSGVO/nDSG) + cookie-einstellungen.html	Compliance-konform
K	Cart-Funktionalität (Schema v2, Backward-compat v1, 7d-expire, MAX_QTY 99)	Persistenz robust

L	Order-Confirmation-Page ( <a href="#">bestellung-bestaetigt.html</a> )	Hero+Next-Steps+Items+Address+Trust+CTA
M	Inventory-Stock-Display (Material-Block, Stock-tier, Notify-when-available)	UX vollständig
N	Search/Filter (Search-Input + Filter-Chips + URL-Sync + 6-field Hay)	Bidirektional sync
O	404 + 500 + Error-Pages (cream BG, Search-Autofocus, Quicklinks, ?msg=URL-Param)	UX-konsistent
P	Performance-Optimierung (Cache-Headers, Preconnect, Lazy-Load, Defer, CSS+JS Minify)	-19% Wire-Bytes vor gzip
Q	Operations-Manual v1.1 + Dashboard-Sync + Manual-Download-Sektion (PBKDF2-PW)	Don-Independence + Audit-Trail
R	Email-Worker-Skelett (CF-Worker, provider-agnostic, Idempotency, AES-Decrypt, 7/7 Tests)	13 Files / 743 LOC ready
S	Material-Hint Don's Formulierung (1428 variants / 349 produkte)	"Materialangabe folgt — für Fragen: info@blackdress.ch"
T	Material-Filter — GESTRICHEN (Don's Entscheidung)	unnötig — 23% default-hint würde unfair filtern
U	Wishlist-Funktionalität (verifiziert)	bereits in K/L/M implementiert (BD.toggleWishlist + wishlist.html)
V	Account-Mock-Skeleton (verifiziert)	bereits implementiert (login/register/konto/passwort-reset, bd_user_v1)
W	Newsletter-Frontend (verifiziert)	bereits implementiert (BD.newsletterSubmit + newsletter-bestaetigt.html)
X	Order-Status-Page Mock-Upgrade	bd_last_order-Lookup + ?nr=URL-Param + dynamic Timeline + Items+Address
Y	Schema.org Phase 2 (160 JSON-LD-Types) + Strict Validation + aggregateRating-Fix	72 Org + 56 Breadcrumb + 30 Article + 1 WebSite + 1 Store, 0 errors
Z	Test-Suite Suite-1 + Suite-9 aktiviert (13 Cases)	13/13 GRÜN nach Don's Disciplined-Debugging-Direktive
Q.1/3	BlackDress-TODO-Cleanup (registers + todos)	1 deleted (material-bulk-fix done), 1 renamed (d1-prod-deploy), 2 added (payrex-integration-worker + email-provider-domain-verification)

Q.2/3 Tag 9 Tagebuch vollständig 40 Highlights, 7 Zusatz-Lessons, 6 Decisions strukturiert

Q.3/3 BlackDress Pages-Index 72 Pages in 11 Kategorien + 9 Speziell-URLs, MD + Dashboard-Sektion

### Pre-Step (vor A):

- og:image → `og-blackdress.jpg` (1200×630, einheitliches Branding alle 72 Pages)
- Material-Schema in `products.json` (1338 extracted + 349 default-hint mit Don's Formulierung)
- Material-Heuristik v2 (extended keywords + HTML-pollution defense)

### Tag-9-Schluss-Inventar:

- 72 HTML-Pages live (42 Top-Level + 30 Blog)
- 18 Skills (~155 KB)
- 13 Crons aktiv
- Brain: 40 Facts / 20 Decisions / 42 Lessons / 236 Tags
- Cost: 0 CHF heute (xai \$0.06 IdeaHunter)

## 11 · Update-Disziplin

Diese Datei wird aktualisiert bei:

- Neuer HTML-Page (Site-Map ergänzen)
- Schema-Änderung (D1 / KV / R2)
- Neuer Drittservice (Sektion 5 ergänzen)
- Neuer Cron (Sektion 6 ergänzen)
- Datenfluss-Änderung (z.B. neuer Payment-Provider neben Payrex)
- Neuer Build-Step → Sektion 10 ergänzen

Brain-Touchpoint: Architektur-Änderungen werden als `decisions` mit Tag `architecture-change` geloggt + diese Datei aktualisiert.

### Cross-Refs:

- `blackdress_zugangs_inventar.md` — Welche Credentials wo
- `blackdress_operations_handbuch.md` — Schritt-für-Schritt Don-Aktionen