

BlackDress - Zugangs-Inventar

Version: 1.2 **Erstellt:** 2026-05-08 **Letzte Aktualisierung:** 2026-05-08 abend (Build-Tag A-Z + Email-Worker + Pages-Index) **Verantwortlich:** Don (alleiniger Inhaber aller Credentials) **Klassifikation:** VERTRAULICH — nicht digital ablegen ohne Verschlüsselung

Wichtig: Diese Liste enthält **KEINE Passwörter im Klartext**. Passwörter werden nur in Don's Passwort-Manager (KeePassXC / 1Password / etc.) gespeichert. "Speicherort" = Eintragsname im Passwort-Manager, **nicht** das Passwort selbst.

So liest du dieses Dokument

Pro Service eine Zeile mit:

- **Service-Name** und **Login-URL**
 - **User / Account-Identifizier**
 - **Speicherort des Passwords** (Don's PW-Manager-Eintrag)
 - **2FA-Status** (aktiv / inaktiv / Recovery-Codes wo)
 - **Was wird damit gemacht** (Use-Case in 1 Satz)
-

A · Hosting & Infrastruktur

Hostinger VPS (KVM-2)

- **URL:** `https://hpanel.hostinger.com`
- **Account-Email:** `<Don ergänzt>`
- **PW-Speicherort:** `<KeePassXC-Eintrag: Hostinger>`
- **2FA:** `<aktiv? Recovery-Codes wo?>`
- **VPS-Server:** `srv1611647 · IP <aus hostinger panel>`
- **Use-Case:** VPS-Reboot, Disk-Resize, Snapshots, IP-Verwaltung

VPS SSH-Root

- **Host:** `<IP des VPS>`
- **User:** `root`
- **Auth:** SSH-Public-Key (kein PW-Login möglich)
- **Private-Key-Speicherort:** `<Don's Local-Maschine: ~/.ssh/id_<name>> + verschlüsseltes Backup <Speicherort>`
- **Use-Case:** Direkter Server-Zugang, Genesis neu starten, Logs lesen

Cloudflare-Account

- **URL:** `https://dash.cloudflare.com`
- **Account-Email:** `<Don ergänzt>`

- **PW-Speicherort:** <KeePassXC-Eintrag: Cloudflare>
- **2FA:** <aktiv? Authenticator-App wo?>
- **Account-ID:** 446df4d003cd8b28ed2dfcdd246d9016 (verifiziert via wrangler)
- **Use-Case:** Pages, DNS, D1, Workers, R2, KV, Email Routing
- **API-Token-Pfad** (auf VPS): /root/.cloudflare (chmod 600, root only)

Cloudflare Pages-Projekte (4 aktiv)

Projekt	Custom-Domain	Pages-URL
blackdress-demo	www.blackdress.ch	blackdress-demo.pages.dev
bot-imperium	(keine)	bot-imperium.pages.dev
knowledge-imperium	(keine)	knowledge-imperium.pages.dev
trading-imperium	(keine)	trading-imperium.pages.dev

B · Domain-Registrar

blackdress.ch

- **Registrar:** Wix.com (NICHT Cloudflare — siehe project_blackdress.md)
- **Registrar-Account:** <Wix-Login-Email>
- **PW-Speicherort:** <KeePassXC-Eintrag: Wix>
- **2FA:** <aktiv?>
- **DNS-Mode:** Wix-DNS-only mit CNAME → CF Pages (Plan B Migration 2026-05-05)
- **Wichtig:** 5 MX-Records (Google Workspace), 4 DKIM-CNAMEs, 2 TXT — diese NICHT ändern ohne Mail-Backup
- **Renewal-Datum:** <Don ergänzt aus Wix-Account>
- **Use-Case:** Domain-Verlängerung, Mail-DNS-Pflege

C · E-Mail

Google Workspace (für blackdress.ch)

- **Admin-URL:** https://admin.google.com
- **Admin-Email:** <Don ergänzt>
- **PW-Speicherort:** <KeePassXC-Eintrag: Google Workspace>
- **2FA:** <aktiv? Recovery wo?>
- **Mail-Adressen aktiv:** info@blackdress.ch, <weitere?>
- **Use-Case:** Kunden-Anfragen, Newsletter-Auswertung, Order-Mail-Forwarding

Email-Sending-Provider (für transaktionale Emails)

- **Aktuell:** Worker-Skelett ready seit Step R (/root/blackdress/workers/email/ — provider-agnostic)

- **Provider-Wahl pendend:** Cloudflare Email-Workers (native, kein Limit) ODER Brevo (free 300/Tag)
- **Email-Worker-Pfad:** `/root/blackdress/workers/email/`
 - `src/index.js` — Entry mit Bearer-Auth + Routing
 - `src/providers/stub.js|brevo.js|cf-email-workers.js` — Adapter (env.EMAIL_PROVIDER wählt)
 - `wrangler.toml` — D1-Binding `blackdress-test`, Production-Override für `blackdress-prod`
- **Required Secrets** (via `wrangler secret put`):
 - `WEBHOOK_AUTH_TOKEN` — Payrexx-Webhook-Auth (generate: `openssl rand -hex 32`)
 - `BLACKDRESS_AES_KEY` — base64 AES-256 für PII-Decrypt (matches checkout-worker)
 - `EMAIL_PROVIDER` — 'stub' (default) | 'brevo' | 'cf-email-workers'
 - `BREVO_API_KEY` — nur falls Provider=brevo (Speicherort: `<KeePassXC>`)
- **Use-Case:** Worker triggered von Payrexx-Webhook bei status=paid/shipped/cancelled → lädt Order aus D1 + decrypted PII + rendert Template + sendet via Provider + loggt in `email_log` (Idempotency)

D · Zahlung

Payrexx (Schweizer Zahlungsanbieter)

- **URL:** `https://payrexx.com`
- **Account-Email:** `<Don ergänzt nach Account-Setup>`
- **PW-Speicherort:** `<KeePassXC-Eintrag: Payrexx>`
- **2FA:** `<bei Setup aktivieren>`
- **API-Keys** (Test + Live):
 - **Test-Key-Speicherort:** `<KeePassXC>` + Cloudflare-Worker-Secret `BLACKDRESS_PAYREXX_TEST_KEY`
 - **Live-Key-Speicherort:** `<KeePassXC>` + Cloudflare-Worker-Secret `BLACKDRESS_PAYREXX_LIVE_KEY`
- **Status:** Account-Setup pendet (TODO `blackdress-payrexx-keys`, deadline 13.5.)
- **Use-Case:** Kreditkarten / TWINT / PayPal / Klarna direkt im Checkout, Refunds, Settlement-Reports

E · Code & Repos

GitHub (falls verwendet)

- **Aktuell:** KEIN Git-Remote für `/root/blackdress` (lokales-only)
- **Falls zukünftig:** Repo-URL + PW-Speicherort hier ergänzen

VPS-File-Struktur (Stand 2026-05-08 nach Step P)

- `/root/blackdress/` — Source der Site (72 HTML + assets + data + lib + d1 + email_templates + manual + test_plans)
- `/root/.blackdress_credentials` — Salt für customer_hash (chmod 600, root only) — Speicherort: nur lokal auf VPS
- `/root/.cloudflare` — CF API Token (chmod 600) — Speicherort: nur lokal auf VPS

OG-Image / Brand-Assets (Step P-Pre-Step, 2026-05-08)

- **OG-Image (Telegram/Social-Vorschau):** `/root/blackdress/assets/og-blackdress.jpg` (1200×630, ~67 KB)
 - Live-URL: `https://www.blackdress.ch/assets/og-blackdress.jpg`
 - Wird von ALLEN 72 HTML-Pages als `og:image` + `twitter:image` referenziert
 - Cache-Header: `max-age=31536000, immutable` (1 Jahr)
 - Build-Source: `/tmp/build_og_image.py` (PIL-Script — falls Re-Build nötig)
 - **Bei Logo-Änderung:** neuer Filename (z.B. `og-blackdress-v2.jpg`) als automatischer Cache-Bust für Telegram (caches per-URL)
- **Logo-Source:** `/root/blackdress/assets/logo.png` (Master-Logo)
- **About-Hero-Image:** `/root/blackdress/assets/about-hero.jpg` (CSS-Background für `about.html` — NICHT mit OG-Image verwechseln)

F · Communication

Telegram-Bot (Genesis ↔ Don)

- **Bot-Name:** `@Genesis1608_Bot`
- **Bot-Token-Speicherort:**
 - VPS: `/root/.claude/channels/telegram/.env`
 - Don's Backup: `<KeePassXC-Eintrag: Telegram Bot Genesis>`
- **Don's Chat-ID:** `6974524199`
- **Use-Case:** Kommunikation Genesis ↔ Don, Bot-Notifications, Cost-Approvals

G · KI / API-Services

Anthropic (Claude)

- **URL:** `https://console.anthropic.com`
- **Account-Email:** `<Don ergänzt>`
- **PW-Speicherort:** `<KeePassXC-Eintrag: Anthropic>`
- **2FA:** `<aktiv?>`
- **Subscription:** Claude Max (\$216.20/Mo, auto-renew 27.05.2026, Mastercard ****6753)
- **API-Key (separater fürs Trading-Bot):** Speicherort `<KeePassXC>` + ENV-Var auf VPS
- **Use-Case:** Genesis-Operations, Sub-Agents, Coding-Builds (all-included via Subscription); separater API-Key nur für Trading-Bot live calls

fal.ai (Bild-Generierung via Genesis-Image)

- **URL:** `https://fal.ai`
- **Account-Email:** `<Don ergänzt>`
- **PW-Speicherort:** `<KeePassXC-Eintrag: fal.ai>`
- **API-Key-Speicherort:** `<KeePassXC>` + `/root/genesis-image/.env`

- **Balance:** ~63 CHF (Stand 2026-05-03)
- **Use-Case:** Hero-Shots, Lifestyle-Bilder, Product-Photography (Pre-Mortem + Cost-Approval-pflichtig per revenue-discipline)

xAI Grok (für Trading-Bot X-Suche)

- **URL:** `https://x.ai`
- **API-Key-Speicherort:** `<KeePassXC>` + ENV-Var auf VPS
- **Budget geladen:** ~\$38
- **Use-Case:** nur Trading-Bot Grok-Fallback bei Nitter-Ausfällen

H · Drittservices (Phase D+, noch nicht aktiv)

Helius (Solana RPC, falls Trading-Bot live)

- **URL:** `https://helius.dev`
- **Status:** Free-Tier-Key benötigt (TODO `helius-api-key`), noch nicht beschafft
- **Wenn beschafft:** Speicherort `<KeePassXC>` + Worker-Secret `HELIUS_API_KEY`

Cloudflare D1 (BlackDress Datenbank)

- **Test-DB:** `blackdress-test` ✓ deployed 2026-05-08 (Step D)
 - **Database-ID:** `42aa3e86-e1de-43be-90a5-26e0748169dd`
 - **Schema deployed:** `/root/blackdress/d1/schema.sql` (10 Tabellen)
 - **Use-Case:** Schema-Validierung, Worker-Connectivity-Tests, Lasttests vor Live
 - **Zugang:** Cloudflare-Dashboard → Workers & Pages → D1 → `blackdress-test` (SQL-Console)
- **Prod-DB:** `blackdress-prod` (noch nicht erstellt — TODO post-Live)
 - **Erstellung:** `wrangler d1 create blackdress-prod` → Schema-Migration via `wrangler d1 execute --file=d1/schema.sql`
 - **Status:** pendet bis Anwaltsdokumente + Payrexx-Setup durch (deadline 13.5.+)

I · Notfall-Recovery

Wenn Don alle Credentials verliert

- **Recovery-Pfad 1:** KeePassXC-Master-Password (single source of truth)
 - **Master-PW-Backup:** `<physisch – Stahl-Plate / 2× Papier an verschiedenen Orten>`
- **Recovery-Pfad 2:** Account-Recovery pro Service via Email + Phone
- **Recovery-Pfad 3:** Server-Side-Backups (Hostinger Snapshots) — sind Don's letzte Verteidigung

Wenn Genesis kompromittiert wird

- Telegram-Bot-Token rotieren (`https://t.me/BotFather` → `/revoke`)
- Cloudflare API-Token rotieren (Dashboard → My Profile → API Tokens)
- Anthropic API-Key rotieren (Console → API Keys)

- Alle Worker-Secrets rotieren (`wrangler secret put`)
-

J · Update-Disziplin

Diese Datei wird aktualisiert:

- Bei jedem **neuen Drittservice** (z.B. wenn Email-Provider live geht)
- Bei **Credential-Rotation** (PW-Wechsel → Speicherort bleibt gleich, evtl. Eintragsname-Update)
- Bei **2FA-Aktivierung/-Deaktivierung**

Genesis aktualisiert das Dokument bei Architektur-Änderungen, Don füllt User/PW-Speicherort selbst aus (Genesis kennt diese Daten nicht und soll sie nicht kennen).

Cross-Refs:

- `blackdress_architektur.md` — Welcher Service macht was im System
- `blackdress_operations_handbuch.md` — Schritt-für-Schritt-Anleitungen