

ARBEITSPROBEN

In mehreren Jahren Plattform- und Produktentwicklung habe ich eine komplette Systemlandschaft rund um Influencer-Marketing, E-Commerce und B2B-Portale aufgebaut und weiterentwickelt.

Die Beispiele unten zeigen zentrale Projekte mit Fokus auf:

- skalierbare Architektur (Microservices, GraphQL-Backbone, Cluster-Datenbanken)
- datengetriebene Funktionen (Suche, Analytics, OLAP, Conversion-Tracking)
- zuverlässigen Betrieb (Monitoring, Backups, SSO, Infrastruktur-Automation)

1. PLATTFORM & ARCHITEKTUR

Plattform-Überblick

- **Systemlandschaft:** rund 38 Anwendungen und Services (Frontends, Backends, Microservices, Infrastruktur).
- **Backbone:** zentrale GraphQL-APIs (GraphQL/GQL) als Datendrehscheibe für Portale, Dashboards und Tools.
- **Kernkomponenten:** SSO/Logins, CDN/Upload, Solr-Suche, Model-Server (ML/OCR), Tracking-Services, Cron-Orchestrierung.
- **Frontends:** Svelte/SvelteKit und Next.js/React, alle konsumieren GraphQL-APIs.
- **Betrieb:** Nagios-Monitoring, Cron-Utils für Jobs (Backups, Crawling, Feeds), Blue-Green-Deployments, Hetzner-Cloud-Infrastruktur.

Microservice-Architektur & Systemlandschaft

- Microservice-Architektur aufgebaut, in der einzelne Funktionsbereiche (Frontends, Backends, ML, CDN, Crawler, SSO, Upload, Tracking) eigenständig laufen.
- Services kommunizieren über APIs hinter Loadbalancern und lassen sich getrennt skalieren.
- Rund 25 Hetzner-Instanzen laufen in einem privaten Netzwerk, das internen Verkehr vom öffentlichen Internet trennt.
- Datenebene: Percona XtraDB Cluster, MySQL/MariaDB, ClickHouse, Solr.
- Infrastruktur: Nginx, PHP-FPM, Node.js-Services, Python-APIs, GlusterFS, CDN-Storage.

Backup-Strategie & Wartung

- Backup-Strategie eingeführt, die Applikationsdaten und Datenbanken automatisch und zeitgesteuert sichert.
- Backups werden lokal erstellt, verschlüsselt und auf separate Backup-Server synchronisiert.
- Regelmäßige Restore-Tests durchgeführt, um Datenintegrität zu prüfen.
- Monitoring (Nagios) nutzt Status-Dateien und eigene Checks, um Probleme früh zu erkennen.
- Microservice-Aufbau ermöglicht parallele Entwicklung und schnelle Rollouts für neue Features und Bugfixes.

Infrastruktur (Kurzüberblick)

- **Server-Setup:** Modul-Installer für PHP/Node/Solr/ClickHouse/GlusterFS.
- **Server-Provisioning:** Hetzner-iaC mit hcloud CLI und cloud-init (Netze, Loadbalancer, IP-Schema).
- **Solr:** mehrkerniges Suchsystem für Portal und Typeahead.
- **Roehrig-Nagios:** Monitoring für 19 Hosts (App, GraphQL, SSO, Upload, DB).
- **Nagios-Plugins:** eigene Checks für Web, HAProxy, RAID, DNS, SSL.
- **Cron-Utils:** Orchestrierung von Backups, Feeds, Crawling, Kampagnenjobs, Sitemaps.

2. BELLISSY COMMERCE & PORTAL

Neuentwicklung Beauty-Portal & Content-Monetarisierungsplattform

Ziel: Bestehendes Portal modernisieren, Performance erhöhen und Content monetarisieren.

- bellissy.de und „Shop the Look“ neu aufgesetzt (Next.js/React, GraphQL).
- Infrastruktur auf Hetzner Cloud migriert (Nginx, Percona XtraDB Cluster, Solr, Loadbalancer).
- CI/CD-Pipelines mit GitLab eingerichtet, Releases automatisiert.
- Ergebnis: laut Sistrix stieg die Sichtbarkeit nach dem Relaunch um ca. 250 %.

Technologie-Stack: Hetzner Cloud, Nginx, React, Next.js, GraphQL, Percona XtraDB Cluster, Solr, Load-Balancing, GitLab CI/CD

Automatisierter Import von Produktfeeds & Verfügbarkeiten

Ziel: Produktdaten verschiedener Shops zentral verwalten und aktuell halten.

- Importpipeline für Shop-Feeds (CSV/JSON/XML) entwickelt, die Daten in MySQL übernimmt.
- Validierungslogik implementiert (Format, Bestand, Preise), um fehlerhafte Produkte früh zu filtern.
- Produktstatus so modelliert, dass Suche, Frontend und Kampagnen direkt darauf zugreifen.

Technologie-Stack: PHP, MySQL

Solr-Suche & erweiterte Suchfunktionen

Ziel: Suche neu aufsetzen und für große Produktmengen optimieren.

- Solr integriert, um Volltextsuche und facettierte Filter (Preis, Marke, Kategorie) zu ermöglichen.
- Suchergebnisseite mit Filtern, Sortierung und Facetten entwickelt.
- Null-Ergebnisseite umgesetzt, die Nutzer auf alternative Kategorien, Marken und Produkte lenkt.

Technologie-Stack: Solr, Typeahead, React

Redaktionelles System & Monetarisierung (CraftCMS, AdSense)

Ziel: Content-Pflege vereinfachen und Werbeumsätze steigern.

- Redaktionelles System mit CraftCMS eingeführt (Ratgeber, Editorials, Landingpages).
- Medienauslieferung und Bildverwaltung über CMS und CDN optimiert.

- Google-AdSense-Anzeigen im Portal integriert und so platziert, dass Sichtbarkeit hoch bleibt, ohne UX zu verschlechtern.

Technologie-Stack: CraftCMS, GraphQL, Google AdSense, React

Preisalarme, Verfügbarkeitsbenachrichtigungen & Newsletter

Ziel: Nutzer binden und Portaltraffic durch Benachrichtigungen steigern.

- Wunschlisten-Funktion und Alarme für Preis- und Verfügbarkeitsänderungen umgesetzt.
- Double-Opt-In-Flow für Newsletter und Benachrichtigungen aufgebaut.
- Mail-Versand über GraphQL-basierte Services und Mailjet angebunden.

Technologie-Stack: GraphQL, Mailjet API, React

Warenkorbpixel & Conversion-Tracking

Ziel: Eigene Conversion-Messung aufbauen und externe Affiliate-Kosten senken.

- Tracking-System entwickelt, das Klicks und Käufe in Partnershops erfasst.
- JavaScript-Snippet bereitgestellt, das Händler in ihre Bestellbestätigungsseiten einbinden.
- Conversions und Warenkorbdetails zentral in der Datenbank gespeichert und ausgewertet.

Technologie-Stack: PHP, Nginx, JavaScript

B2B-Portal laserschneiden24.de

Ziel: Angebots- und Bestellprozess für Blechzuschnitte digitalisieren.

- HubSpot-Integration gebaut, um Kundendaten automatisch ins Portal zu übernehmen.
- Upload und Verarbeitung von STP- und DXF-Dateien implementiert, inklusive automatischem „Unfolding“ technischer Details.
- Angebotserstellung und Auftragsverwaltung mit Statusverfolgung im Portal umgesetzt.

Technologie-Stack: PHP, Nginx, HubSpot, Mailjet API, CDN, WiCam API, STP, DXF, Hetzner Cloud

3. CAMPAIGN MANAGEMENT & ANALYTICS

Campaign Self Service (CSS) & Terminplanung

Ziel: Kunden befähigen, Kampagnen selbst zu planen und zu steuern.

- Self-Service-Tool entwickelt, mit dem Kunden Termine planen, Briefings erstellen und Kampagnen überwachen.
- Workflow aufgebaut, der passende Influencer vorschlägt und Kampagnenerfolg über Shortcodes und Warenkorbpixel misst.
- Kalenderansicht mit FullCalendar integriert.

Technologie-Stack: Svelte, FullCalendar

Performance-Tracking für Kundenkampagnen

Ziel: Kampagnen-Performance pro Kunde und Influencer sichtbar machen.

- Shortlink-System gebaut, das auf beliebige Ziel-URLs weiterleitet und Seitenaufrufe als Conversions erfasst.
- Individuelle Shortlinks für Influencer und Kampagnen generiert.
- Datenmodell so gestaltet, dass Attribution und Auswertungen pro Kunde, Kampagne, Link und Influencer möglich sind.

GUI für Kampagnenauswertung (ClickHouse-Reporting)

Ziel: Conversion-Daten schnell und flexibel auswerten.

- Conversion-Daten von MySQL auf ClickHouse migriert, um Abfragen stark zu beschleunigen.
- Reporting-Interface in Svelte entwickelt, mit getrennten Ansichten für Kunden und Customer-Success-Team.
- CSV-Exports integriert, damit Teams Daten in BI-Tools weiterverarbeiten können.

Technologie-Stack: ClickHouse, Svelte, GraphQL, CSV

Aufbau einer Beauty-Community

Ziel: Community-Funktionen für Mitglieder und Influencer einführen.

- Tokenbasiertes Login-System für Mitglieder und Influencer entwickelt.
- Wunschliste ausgebaut: triggert automatisch Preis- und Verfügbarkeitsalarme.
- Social-Features implementiert: Nutzer können Beiträge liken und Influencern folgen.

Technologie-Stack: PHP, GraphQL, MySQL, Mailjet API, React

4. IDENTITY, MESSAGING & CRM

Single Sign-On (SSO) für Plattform-Ökosystem

Ziel: Zentrale Anmeldung für alle Anwendungen.

- Passwortloses SSO-System entwickelt: Login via One-Time-Code per E-Mail statt Passwort.
- Sessions per Cookie verwaltet, inklusive CSRF-Schutz und Rollenmodell pro Anwendung.
- Kundendaten mit HubSpot synchronisiert, um Benutzerverwaltung zu automatisieren.

Technologie-Stack: PHP, Percona XtraDB Cluster, Nginx, Mailjet API, HubSpot, Hetzner Cloud

Mailgun-Integration als E-Mail-Microservice

Ziel: E-Mail-Versand zentralisieren und automatisieren.

- Mailgun als zentralen E-Mail-Service für Newsletter und transaktionale Mails angebunden.
- Templates serverseitig gerendert, Double-Opt-In und Abmelde-Flows umgesetzt.
- Eigene API-Endpoints geschaffen, um Geschäftslogik (Alerts, Kampagnen, Onboarding) mit dem Mailing-System zu verbinden.

Technologie-Stack: Mailgun API, PHP, Nginx, cron, Hetzner Cloud

WhatsApp-Chat-Gateway

Ziel: WhatsApp-Kommunikation in CRM-Prozesse integrieren.

- Node.js/Express-Server als Brücke zur WhatsApp Business Cloud API implementiert.
- Eingehende und ausgehende Nachrichten (Text und Medien) verarbeitet und Zustellstatus in Echtzeit protokolliert.
- Webhook-Verifikation und Signaturprüfung für sichere Anbindung umgesetzt.

Technologie-Stack: Node.js, Express, WhatsApp API

5. DATEN, ML-SERVICES & CRAWLING

Model-Server für ML-APIs

Ziel: Machine-Learning-Modelle zentral bereitstellen.

- Python-basierten Model-Server gebaut, der mehrere ML-Modelle als APIs im privaten Netzwerk bereitstellt.
- Gunicorn-Setup mit eigener Middleware implementiert; Modelle als Read-Only-Strukturen im RAM gehalten, um Antwortzeiten niedrig zu halten.
- OCR- und Analyse-Endpunkte zum Beispiel für Instagram-Story-Insights integriert.

Technologie-Stack: Python, Gunicorn, OpenCV, Tesseract, Hetzner Cloud

Crawler & Content-Recording

Ziel: Social-Media-Daten für Analysen und Kampagnen automatisiert sammeln.

- Browser-basierte Crawler mit ferngesteuertem Chrome (CDP) entwickelt.
- Mobile-App-basierte Crawler via Appium aufgebaut, um Profile in der Instagram-App automatisiert zu durchlaufen.
- Daten in eigene Datenbanken und CRM-Systeme (HubSpot) übernommen.

Technologie-Stack: Docker, Node.js, Chrome Remote Interface, Appium, HubSpot

Covid-Preisvergleichsportal für Testzentren

Ziel: Übersicht über COVID-Testzentren und Preise in Deutschland schaffen.

- Crawling-Instanz in der Cloud aufgebaut, die regelmäßig 12+ Anbieterwebseiten ausliest.
- Postprocessing implementiert: neue Städte erkennen, Adressen geokodieren, Daten bereinigen.
- Tracking via Google Tag Manager optimiert und Self-Service-API für Teststellenanbieter bereitgestellt.

Technologie-Stack: Node.js, Chrome Remote Interface, Nginx, PHP, Solr, Google Tag Manager, Cookiebot, Hetzner Cloud

6. WEITERE PROJEKTE & SPEZIALLÖSUNGEN

Content Delivery Network (CDN) & Upload-Service

Ziel: Medien performant und kontrolliert ausliefern.

- Eigenes CDN mit Upload- und Download-Service entwickelt.
- Rechteverwaltung pro Anwendung und Asset-Typ umgesetzt (Avatare, Kampagnenbilder, Briefings etc.).
- Bildpipeline mit automatischem Resizing, Kompression und content-addressed Storage (SHA256) implementiert.
- Nginx X-Accel-Redirect genutzt, um Dateien effizient und sicher auszuliefern.

Technologie-Stack: PHP, Nginx, SSO, GlusterFS, Hetzner Cloud, ImageMagick

Villagari – Ferienhaus-Portal

Ziel: Luxus-Ferienhaus-Webauftritt mit schnellem, statischem Frontend und sicherem CMS realisieren.

- WordPress-basierte Bedrock-Installation als CMS-Backend aufgesetzt.

- Statische HTML-Generierung mit getrenntem Frontend umgesetzt (schnelles, CDN-freundliches Hosting).
- Mehrsprachige Inhalte, saisonale Preislogik und strukturierte Daten modelliert.
- Git-basierte Deployment- und Static-Build-Prozesse mit Rollback-Optionen entwickelt.

Technologie-Stack: PHP, WordPress/Bedrock, MySQL, Nginx/PHP-FPM, wp2static, Git

7. SYSTEMÜBERSICHT NACH DOMÄNE (AUSZUG)

Frontend-Anwendungen (Auswahl)

- **Campaign Self Service:** Self-Service-Tool für Kampagnen (SvelteKit, HubSpot, Mailgun, JSON-Storage).
- **Portal-Next:** E-Commerce-Portal mit SSR für bellissy.de (Next.js, GraphQL).
- **Campaign GUI:** Echtzeit-Analytics für Kampagnen (SvelteKit, ClickHouse-Daten).
- **Client Campaign GUI:** Kunden-Dashboards mit CSV-Export.
- **Campaign Control:** Admin-Tool für Kampagnen, Shortlinks, Influencer-Zuordnung.
- **GetInspired (STL):** Creator-Portal mit Rollen, CSRF-Schutz und IBAN-Handling.
- **Roehrig-Frontend:** B2B-Portal mit 3D-Preview und SSO.
- **Insights:** Upload-UI für Instagram-Insights mit OCR-Auswertung.
- **Price-Check-Frontend:** Portal zur Testzentren-Suche (Solr, Mapbox).
- **Villagari:** WordPress-basiertes Vermietungs-CMS με statischer Auslieferung.

Backend & APIs (Auswahl)

- **GraphQL:** Haupt-API für Produkte, Kampagnen, Influencer, Conversions (MySQL, Solr, ClickHouse).
- **GQL:** Kampagnen-spezifische GraphQL-Instanz mit RBAC.
- **Old-Backend:** Symfony-basierter Altbestand für Tracking und Katalog.
- **Roehrig-Backend:** GraphQL-API für Angebote, Aufträge, Kunden.
- **Bellissy Framework:** eigenes PHP-MVC-Framework (DB-Abstraktion, Validator, CSRF).

Microservices (Auswahl)

- **Model-Server:** ML/OCR-Service für Kategorisierung, Relevanz und Story-Auswertung.
- **Bellissy CDN:** Upload- und Image-Pipeline mit SSO/RBAC und X-Accel-Redirect.
- **Roehrig-Upload:** Upload-Service mit Queue, Rollen und Checksums.
- **SSO / Login / Roehrig-Login:** passwortloses Login mit Token, CSRF, Rate-Limiting, HubSpot-Sync.
- **HubSpot:** CRM-Sync plus Social-Media-Crawler.
- **WhatsApp:** Cloud-API-Bridge mit Status-Tracking und 24h-Fenster-Logik.
- **Feed-Tools:** 52 Lieferantenfeeds → DB + Solr.
- **COVID-Crawler:** Testzentren-Crawling und Geocoding.
- **Image-Tools:** Bild-Pipeline für Produktfotos.
- **Mailgun:** Newsletter- und Transaktionsmails als Microservice.
- **Shop-Redirect / Shopin / Tracking-Pixel:** Shortlinks, Attribution, serverseitiges Conversion-Tracking.

Infrastruktur

- **Server-Setup:** Modul-Installer für PHP/Node/Solr/ClickHouse/GlusterFS.
- **Server-Provisioning:** Hetzner-laC mit hcloud CLI und cloud-init.
- **Solr:** mehrkerniges Suchsystem (Portal + Typeahead).
- **Roehrig-Nagios:** Monitoring für 19 Hosts.
- **Nagios-Plugins:** eigene Checks für Web, HAProxy, RAID, DNS, SSL.
- **Cron-Utils:** Orchestrierung von Backups, Feeds, Crawling, Kampagnenjobs, Sitemaps.