Engineer

Portfolio

Ingenieurleistungen mit Beispielen umgesetzter Arbeiten.

Engineer ApS wurde 2025 eingetragen. Die meisten hier gezeigten Arbeiten hat unsere Gründerin zuvor in Kundenteams umgesetzt; einige hat das Unternehmen selbst erbracht, andere die Gründerin seitdem in Anstellungsverhältnissen. Dieselben Hände und dieselben Standards — wir benennen den jeweiligen Kontext, statt die Grenzen zu verwischen.

Entdecken Sie die Arbeiten **in ihrem Kontext** — die Entscheidungen, die Umsetzung und die dokumentierten Ergebnisse.

Gesamten Katalog ansehen

Erfahrung finden, die sich nutzen lässt

Nutzen Sie diese Arbeitsnachweise, um Erfahrung zu finden, die für Ihr Projekt relevant ist.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Entscheidungen hinter den Arbeiten lesen

Die Einträge beschreiben umgesetzte Arbeiten über verschiedene Leistungen hinweg. Lesen Sie eine Rückschau, um die Situation und Entscheidungen dahinter zu verstehen.

Fallstudien Kontakt aufnehmen →

Diese Namen **ziehen sich durch die folgenden Arbeiten** — nichts Exotisches, das Ihr Team übernehmen müsste.

Python PostgreSQL TypeScript React GitHub Docker

Gesamten Technologie-Stack ansehen

Nach Kategorie stöbern

SOFTWAREENTWICKLUNG

Backend-Entwicklung Frontend-Entwicklung Full-Stack-Entwicklung APIs und Integration Webentwicklung

Designsysteme und UI · UX / UI-Design · Tests und Qualitätssicherung · Performanceoptimierung · Python · Internationalisierung

DATEN & DATENBANKEN

Datenbanken · PostgreSQL · SQL · Data Engineering · Datenpipelines (ETL/ELT) · Datenanalyse · Data Governance · Data Warehousing · GIS / Geodaten

CLOUD, INFRASTRUKTUR & DEVOPS

Cloud · Container (Docker/Kubernetes) · DevOps · Automatisierung und CI/CD · Infrastruktur · Linux und Server · Systemadministration · Netzwerke und VPN · Monitoring und Observability · Zuverlässigkeit und Sicherungen · Sicherheit · Plattformarchitektur · Migrationen und Modernisierung

BERATUNG & FÜHRUNG

Technische Leitung · Teamführung · Mentoring und Coaching · Agile Methoden und Scrum · Projektmanagement · Produkt und Anforderungen · Stakeholder und Berichterstattung · Dokumentation · Lösungsarchitektur · Marke und Marketing

Alles, was folgt, **wurde umgesetzt** — nicht vorgeschlagen, nicht geplant.

Jeder Eintrag benennt, was die Arbeit verändert hat, und verlinkt die Leistungen, die sie belegt.

Was wir anbieten Was es kostet →

{ Softwareentwicklung }

Plattform- und Lösungsarchitektur

Klären, wie das Ganze aufgebaut werden soll

Die Struktur eines Systems wird zuerst auf dem Papier durchdacht, damit die Teile zusammenpassen und die Kosten feststehen, bevor die erste Zeile Code entsteht.

Swift-6-Nebenläufigkeit über C abgesichert

Vollständige strikte Nebenläufigkeit von Swift 6 ohne eigene Actors eingeführt und eine blockierende C-Ereignisschleife über genau einen Produzenten, einen Verbraucher und eine Reihenfolge an den Main Actor angebunden, nachdem ein Task je Ereignis die für die Oberfläche nötige Reihenfolge verloren hatte.

Eine begrenzte Ereignispipeline für eine macOS-App

Ereignispuffer und Protokollierung nach einem dokumentierten macOS-Vorfall mit 41 MB Wachstum pro Sekunde und 111 GB komprimierten Speicherseiten auf einem Rechner mit 36 GB begrenzt; ein dokumentierter vollständiger Testlauf reduzierte Kernprotokollzeilen von 610.996 auf 1.411.

Nativen macOS-Client in Swift 6 entwickelt

Einen nativen macOS-Messaging-Client in Swift 6 und SwiftUI entwickelt — 11.141 Zeilen in 53 Dateien — auf der C-Schnittstelle eines Rust-Kerns, der aus einer festgelegten Revision als statisches Archiv eingebunden wird.

QR-Encoder mit 522 Zeilen eingebunden

Einen QR-Encoder mit Reed-Solomon über GF(256), festem Modullayout und acht Maskenmustern in 522 Zeilen eingebunden, statt eine dritte Laufzeitabhängigkeit einzuführen, und durch Rücklesen der fertigen Matrix mit einem unabhängig geschriebenen Decoder geprüft.

Datenbankgestützten Lebenslaufgenerator aufgebaut

Einen datenbankgestützten Generator für Lebenslauf, Referenzen, Portfolio und Anschreiben in Python entwickelt — 41 Module, 10.580 Zeilen — der sechs Ausgabeformate aus einer SQLite-Quelle mit 23 Tabellen erzeugt, aufgebaut durch eine idempotente Pipeline mit 19 Schritten.

Öffentliche Dienste auf einem kleinen Server

Die öffentlichen Unternehmensdienste auf einem Single-Core-Host mit 512 MB gehostet — Git-Forge, Webserver für sieben Domains, Tor, zwei Server für alternative Protokolle und Angriffssperren — mit 464 MB nutzbarem Arbeitsspeicher als Entwurfsgrenze; Sicherungsautomatisierung ist eine getrennte Implementierung.

Die gesamte Kerninfrastruktur neu aufgebaut

Als eine der ersten Mitarbeiterinnen die gesamte Kerninfrastruktur und die unterstützenden Prozesse eines Green-Energy-SaaS-Start-ups von Grund auf entworfen und aufgebaut und damit die Grundlage für schnelles Wachstum gelegt.

Infrastruktur entwickelt, die Jahre später läuft

Neue Infrastrukturfunktionen für interne und externe Systeme geplant und umgesetzt und dabei langlebige Lösungen geschaffen, die Jahre später mit minimalen Änderungen weiterlaufen.

Technische Grundlagen der Plattform festgelegt

Die grundlegenden Plattformentscheidungen in den Wochen nach der Eröffnung des Repositorys im November 2025 festgelegt — Schichtenarchitektur, datenbankzentrierter Datenzugriff und Warnungsfreiheit — die neun Monate später weiterhin gelten.

348 Tabellen in 34 Postgres-Schemata modelliert

Die maritime Domäne in 348 normalisierten Tabellen über 34 PostgreSQL-Schemata modelliert — Fachkräfte, Unternehmen, Schiffe, Stellen, Bewertungen und weitere Bereiche — mit SMALLINT-Nachschlagetabellen und UUID-v7-Schlüsseln.

PostgreSQL-Datenschicht mit Funktionen entworfen

Eine PostgreSQL-Datenschicht nach dem Function-first-Prinzip entworfen — 1.275 gespeicherte Funktionen in 34 Schemata — sodass jeder Lese- und Schreibzugriff über eine gezielt berechtigbare Funktion statt über eine Tabelle erfolgt.

API auf Huma v2 migriert, 100 % Übereinstimmung

Die HTTP-API von Fiber auf Huma v2 migriert — 649 Pfade und 760 Operationen — und eine vollständige Übereinstimmung (100 %) zwischen den registrierten Serverrouten und der veröffentlichten OpenAPI-Beschreibung erreicht und beibehalten.

Organisationswechsel für Nutzer entwickelt

Einen Wechsel des Organisationskontexts mit clientseitigem localStorage und serverseitiger Cookie-Spiegelung entworfen, sodass Nutzer im Namen verwalteter Organisationen handeln können, während das Prinzip minimaler Berechtigungen gilt.

JSON direkt von Postgres an die API geleitet

Eine JSON-Passthrough-Architektur entworfen, in der PostgreSQL-Funktionen vollständiges JSON zurückgeben, das die Go-API unverändert weiterreicht; damit entfällt die zwischenzeitliche Deserialisierung und das Frontend bleibt von Schemaänderungen entkoppelt.

Eine maritime Plattform mit klaren Systemgrenzen

Eine geschichtete maritime Plattform mit getrenntem Next.js-PWA-Frontend, Go-API (Huma/Fiber) und PostgreSQL-Funktionsschicht entworfen und sämtliche Geschäftslogik in der Datenbank gehalten.

Infrastruktur vereinfacht und AWS verlassen

Die Infrastruktur eines internationalen Kunden überarbeitet, ungenutzte Dienste entfernt, Kapazitäten angepasst und Workloads von AWS weg verlagert. Die Gründerin berichtet von einer Senkung der Cloudkosten um den Faktor 10 ohne Produktivitätsverlust.

Ein Kartenprodukt durchgängig verantwortet

Infrastruktur, Datenverarbeitung und eine Kartenanwendung über 2 Jahre konzipiert, entwickelt und betreut, einschließlich Softwareauslieferung und betrieblicher Wartung.

Abläufe überarbeitet und 8.000 Stunden eingespart

Interne Abläufe überarbeitet und durch bessere Softwarearchitektur, Systeme und Terminplanung 8.000 Stunden eingespart.

GIS-Kartenprodukt zu 10-fachem Umsatz geführt

Die Softwareentwicklung einer GIS-Kartenanwendung geleitet, den Umsatz um das 10-Fache gesteigert und das Produkt als zentralen Datenbestand positioniert.

Ein Infrastrukturplan für 14 Abteilungen

Ein umfassendes Infrastrukturrahmenwerk für ein Forschungsinstitut mit 14 Abteilungen entworfen, mit flexiblen Modulen, einheitlichen Datenpipelines und strukturierten Supportstrategien für eine langfristige Nutzung.

{ Softwareentwicklung }

Backend- und API-Entwicklung

Der Antrieb hinter Ihrer App oder Website

Der Teil, den Nutzer nie sehen: wo die Daten liegen, welche Regeln gelten und wie andere Systeme darauf zugreifen dürfen.

Die ungenutzte Hälfte des Kerns erschlossen

Die zuvor ungenutzte Hälfte des Messaging-Kerns erschlossen — Sicherungsübertragung, verschwindende Nachrichten, Nachrichtenbearbeitung und erneutes Senden, verifizierte Einladungen, Proxys und Verschlüsselungsrichtlinien — und jeden Test ohne Mocks gegen die echte Bibliothek ausgeführt.

Parser zur Prüfung aller FFI-Aufrufe geschrieben

Einen Parser geschrieben, der den echten C-Header mit 7.308 Zeilen liest und jede Aufrufstelle, jede Enum-Konstante sowie final für jede pointerbesitzende Klasse prüft, nachdem ein handgeschriebener Platzhalterheader Aufrufe dreier entfernter Funktionen kompilieren und linken ließ, die zur Laufzeit abstürzten.

Jeden Text in einen Inhaltsbaum verlagert

Jede sichtbare Zeichenfolge aus Python in einen Inhaltsbaum mit 874 Dateien in drei Sprachen verlagert, nachdem unerkannte tote Übersetzungen und eine stillschweigend nur ein Drittel der Leistungen bewertende Prüfung gefunden worden waren.

Jede Lint-Regel als Fehler aktiviert

Sämtliche Regeln des Python-Linters als Fehler aktiviert und 1.815 Befunde bis auf null abgearbeitet; jede der wenigen Ausnahmen schriftlich begründet und zwei davon durch eine Prüfung statt nur einen Kommentar abgesichert.

Generator mit 981 Tests und 92 % Abdeckung

Den Generator mit 981 Testfällen und mindestens 92 % Zweigabdeckung geprüft, Warnungen als Fehler behandelt und Idempotenz durch zwei vollständige Pipeline-Durchläufe aus einer leeren Datei nachgewiesen, wobei der zweite nichts ändern darf.

Stellenmarkt und Seefahrtskarriere entwickelt

Arbeitsbereiche für Personalgewinnung und die Laufbahn von Seeleuten entwickelt; der dokumentierte Implementierungsstand umfasst 151 gespeicherte Funktionen in 48 Tabellen für Stellen, Bewerbungen, Zertifikate, Rangentwicklung und Seefahrtzeit.

Soziale Ebene mit Beiträgen und Gruppen aufgebaut

Die soziale Ebene der Plattform entwickelt — Beiträge, Feed, Gruppen, Erwähnungen, Follower-Graph und eine Übersicht anstehender Anlässe — auf derselben funktionszentrierten Datenschicht wie das übrige Produkt.

Loyalitätskonto, Ligen und Abzeichen entwickelt

Das Loyalitäts- und Reputationssystem entwickelt — 67 Funktionen über ein Hauptbuch mit 31 Tabellen, mit Ligen, Abzeichen und einem Prämienshop — und den Kontostand durch eine Zeilensperre gegen doppelte Ausgaben abgesichert.

Bei Fehlern sperrenden Missbrauchsschutz gebaut

Bei Fehlern sperrende Missbrauchsschutzmechanismen aufgebaut — 22 Redis-gestützte Ratenbegrenzer, Cloudflare Turnstile, Anfrage-Idempotenz und eine Ursprungssperre — damit die Plattform Bots und Überlastungsangriffe abweist, statt ihren Aufrufern zu vertrauen.

Eigene Fehlerüberwachung und Tracing entwickelt

Eigene Fehlerüberwachung und OpenTelemetry-Tracing entwickelt statt eingekauft — Bereinigung von Nutzdaten, Erkennung von Fehlerspitzen und Regressionen, Symbolauflösung und synthetisches Lebenszeichen — mit 11 Betriebsansichten.

Langsame Aufgaben in River-Jobqueue verlagert

Langsame Verarbeitung aus dem Anfragepfad in eine River-Jobqueue verlagert — 15 Worker-Module, 8 geplante Aufgaben und 20 pg_cron-Jobs — sodass Anfragen zurückkehren, während die zugehörige Arbeit weiterläuft.

Stripe- und In-App-Kaufberechtigungen entwickelt

Die Zahlungs- und Berechtigungsschicht aufgebaut — Stripe sowie In-App-Käufe von Apple und Google — und Verzeichnis, Suche und Export durch ein serverseitig geprüftes Zugriffsmodell mit 11 Tabellen abgesichert.

Testsuite für Go, SQL und Browser aufgebaut

Eine mehrschichtige automatisierte Testsuite aufgebaut — 981 Go-Tests, 543 Frontend- und Browserspezifikationen, 494 verhaltensprüfende SQL-Tests — mit Mutationstests, eigenschaftsbasierten Tests und einer verbindlichen Barrierefreiheitsprüfung.

Django-Apps für IPTV entwickelt und gewartet

Django-Webanwendungen für die IPTV-Plattform entwickelt und gewartet, neue Funktionen ausgeliefert sowie Performance und Stabilität verbessert.

Anwendung mit Nonce-basierter CSP gehärtet

Die Anwendung mit Nonce-basierter CSP, HSTS, SameSite-Cookies, Datenbankrollen mit minimalen Berechtigungen und erneuten serverseitigen Zugangsprüfungen gehärtet.

Kataloggesteuerten Deep-Merge umgesetzt

Eine kataloggesteuerte tiefe Zusammenführung gespeicherter JSON-Einstellungen umgesetzt, die Abstürze durch fehlende Schlüssel bei der Weiterentwicklung des Schemas verhindert.

E-Mail mit Failover und Zustellprotokoll gebaut

E-Mail als Plattformfunktion entwickelt — drei Anbieter mit Ausfallsicherung, Zustell-Webhooks, Versand- und Zustellprotokollen, Vorlagen und Kampagnen — hinter einer Startprüfung, die ohne eingerichteten Anbieter den Systemstart verhindert.

Den API-Vertrag aus der Datenbank generiert

Den API-Vertrag von der Datenbank aus nach außen generiert — OpenAPI, einen typisierten TypeScript-Client mit 44.076 Zeilen, 61 Mock-Handler und die von der UI durchgesetzten Grenzen — mit einer Prüfung an jedem Übergang, die Abweichungen zurückweist.

Ein Datenanalyse-Berichtssystem entwickelt

Ein Berichtssystem für Datenanalysen entwickelt und den quartalsweisen Softwareumsatz durch Python-basierte PDF-Berichte um 400 % erhöht.

30+ GIS-Projekte von Entwicklung bis Support

Entwicklung, Bereitstellung und Support von über 30 GIS-Projekten geleitet und dabei Fachwissen zu PostgreSQL, Bash, Python, JavaScript, GDAL, ArcGIS, PostGIS und Mapbox eingesetzt.

Full-Stack-GIS-Kartenentwicklung geleitet

Die Full-Stack-Entwicklung einer GIS-Kartenanwendung mit PostgreSQL, Mapbox, ReactJS und NodeJS geleitet und eine integrierte Lösung umgesetzt.

8 Power-BI-Projekte mit Handbüchern umgesetzt

8 Power-BI-Projekte mit umfassenden Handbüchern umgesetzt und Microsoft Power BI mit einer NodeJS-API und Python FastAPI für wirksame Datenanalyse und Visualisierung integriert.

6 ETL/ELT-Pipelines entworfen und betrieben

6 ETL/ELT-Pipelines mit Google BigQuery, MSSQL, PostgreSQL, Shell-Skripten, PL/pgSQL und Transact-SQL entworfen, umgesetzt und administriert sowie Daten für eine effiziente Verarbeitung durch die Python-API integriert.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Swift-6-Nebenläufigkeit über C abgesichert · QR-Encoder mit 522 Zeilen eingebunden · Datenbankgestützten Lebenslaufgenerator aufgebaut · Technische Grundlagen der Plattform festgelegt · PostgreSQL-Datenschicht mit Funktionen entworfen · API auf Huma v2 migriert, 100 % Übereinstimmung · Organisationswechsel für Nutzer entwickelt · JSON direkt von Postgres an die API geleitet · Eine maritime Plattform mit klaren Systemgrenzen

{ Softwareentwicklung }

Frontend-Entwicklung

Die Oberflächen, mit denen Menschen arbeiten

Oberflächen, die schnell laden, auf dem Smartphone funktionieren und sich in allen Browsern Ihrer Kunden gleich verhalten.

macOS-Designsystem mit 48 Tokens aufgebaut

Ein Designsystem mit 48 Tokens auf dem plattformeigenen Glasmaterial aufgebaut und einen Linter geschrieben, der unbenannte Zahlenwerte, fest codierte Schriftgrößen und Animationen zurückweist, die die Einstellung für reduzierte Bewegung ignorieren.

Alle 17 Symbole für Screenreader benannt

Jedes ausschließlich durch ein Symbol dargestellte Bedienelement für Screenreader benannt, nachdem die Sendeschaltfläche als "Pfeil nach oben im Kreis, Schaltfläche" angekündigt worden war, und einen Linter geschrieben, der die Beschriftung innerhalb von acht Zeilen um das Symbol verlangt.

Qualitätsprüfung mit 10.242 Zeilen selbst geprüft

10.242 Zeilen JavaScript der Qualitätsprüfung unter Formatter und Linter gebracht, nachdem sie sich als größter und zugleich ungeprüfter Codebestand des Repositorys erwiesen hatten; 13 Befunde behoben und keinen unterdrückt.

Stylesheet von 87 KB auf 31 KB verkleinert

Das Stylesheet-Bündel durch das Auslagern einer Base64-Schrift von 87 KB auf 31 KB verkleinert und 756 KB in 22 bei jeder Bereitstellung veröffentlichten, aber nirgends referenzierten Dateien entfernt.

Druckstylesheet mit 637 Zeilen entwickelt

Ein Druckstylesheet mit 637 Zeilen anhand von vier dokumentierten Verhaltensweisen von Rendering-Engines entwickelt, einschließlich eingebundener Überschriften, die bei Lesern mit dunkler Browsereinstellung weiß auf weiß gedruckt wurden.

Reduzierte Bewegung wirksam umgesetzt

Die Einstellung für reduzierte Bewegung wirksam gemacht, nachdem elf Selektoren darunter weiterhin animierten, weil eine universelle transition-none-Regel in der Spezifität gegen jede Regel mit einer Klasse verliert.

Fehlende Schicht im hellen Farbschema gefunden

Durch die Prüfung gleicher Schichtanzahlen auf allen mehrschichtigen Oberflächen beider Farbschemata festgestellt, dass dem hellen Farbschema seit seiner Entstehung eine ganzseitige Überlagerungsschicht fehlte.

Farbsystem auf 28 Farben begrenzt

Das Farbsystem auf 28 dokumentierte Farben begrenzt und einen Linter entwickelt, der verwendete, aber undokumentierte, dokumentierte, aber ungenutzte, falsch gemessene, als Literal neu geschriebene oder einer bestehenden Farbe bis auf einen Wahrnehmungsabstand von 0,02 ähnliche Werte zurückweist.

Barrieren, die das Prüfwerkzeug übersieht

11 Barrierefreiheitsfehler gefunden und behoben, die der Browser als unbedenklich gemeldet hatte — acht trotz pointer-events weiter per Tab erreichbare Footerlinks, eine Scroll-Timeline, die das Kolophon auf drei Seiten abschnitt, und eine Regelengine, die nur 70 ihrer 105 Regeln ausführte.

Barrierefreiheitsprüfungen auf 32 Seiten in beiden Farbschemata

Prüfungen für WCAG 2.2 Level AA auf 32 repräsentativen Seiten in beiden Farbschemata umgesetzt, mit Konformitätsnachweis, einem geprüften AAA-Kriterium und ausdrücklich benanntem Teilergebnis für Kriterium 1.4.13.

27 Schriftgrößen durch 6-stufige Skala ersetzt

27 tatsächlich dargestellte Schriftgrößen mit teils nur 0,6 % Abstand durch eine sechsstufige Skala im Verhältnis einer großen Terz ersetzt und den Linter geschrieben, der eine achtundzwanzigste Größe zurückweist.

Ein Sicherheitsnetz, das keine halbe Stunde mehr kostet

Die browsergestützte Qualitätsprüfung der Website von 1.636 auf 615 Sekunden verkürzt, indem die Prüfungen in einem auf vier parallele Ausführungen begrenzten Worker-Pool nach längster Laufzeit zuerst eingeplant werden, nachdem die alphabetische Reihenfolge 320 statt 224 Sekunden benötigt hatte.

Hugo-Website mehrsprachig in 3 Sprachen aufgebaut

Eine mehrsprachige statische Hugo-Website mit 108 Vorlagendateien einschließlich 53 Partials aufgebaut, die jede Seite aus einem Inhaltsbaum in drei Sprachen und vier Darstellungsformen veröffentlicht, zusätzlich auf sieben spezialisierten Subdomains aus demselben Baum.

Verwaltungs-Backend der Plattform entwickelt

Das Verwaltungs-Backend für den Betrieb entwickelt — rund 40 Administrationsrouten und 128 Komponenten für Besitzansprüche, Moderation, Funktionsschalter, Cache und Diagnostik — damit sich die Plattform ohne Datenbankzugriff betreiben lässt.

Hugo-Website betreut und Produkt-UI verbessert

Die bestehende statische Hugo-Website gepflegt und erweitert sowie UI/UX-Verbesserungen zum zentralen Asset-Management-Produkt beigetragen.

Englisch-dänisches i18n-System aufgebaut

Ein englisch-dänisches Internationalisierungssystem mit 12.027 Nachrichten je Sprachversion in 458 Namespace-Dateien aufgebaut, mit per Linter durchgesetztem Wortschatz und einer Grenze von 400 Zeilen je Datei.

UI und UX des Produkts durchgängig gestaltet

UI und UX des Produkts durchgängig gestaltet — Verzeichnisraster, kombinierte Karten- und Tabellenansichten, laufende Anforderungsprüfungen und Breadcrumb-Navigation.

iOS-26-'Liquid Glass'-Designsystem entworfen

Ein iOS-26-Designsystem im Stil von 'Liquid Glass' und ein verbindliches Komponentenverzeichnis entworfen, das Lint-Regeln gegen abweichende UI-Umsetzungen absichern.

Progressive Web App vollständig unterstützt

Vollständige Progressive-Web-App-Unterstützung umgesetzt — installierbar und offlinefähig — mit Workbox-Laufzeitcaching über next-pwa.

Next.js-16-Frontend: SSR, SSG und CSR entwickelt

Das Next.js-16-Frontend mit bewusst gewählten SSR-, SSG- und CSR-Strategien sowie einer wiederverwendbaren Factory für vorab geladene Serverseiten entwickelt, die die N+1-Abfragekaskade auf jeder umgestellten authentifizierten Seite beseitigt.

30 Websites von Photoshop in HTML umgesetzt

30 Websites gestaltet und durch die Umsetzung von Photoshop-Entwürfen in HTML individuelle, flexible Lösungen geschaffen.

100 Webanwendungen mit HTML/HTML5 entwickelt

100 Webanwendungen mit HTML/HTML5, CSS/SCSS, Django, WordPress und Joomla entwickelt und eine vielseitige Online-Präsenz ermöglicht.

An der Gestaltung von Oberflächen mitgewirkt

An der Gestaltung von Benutzeroberflächen mitgewirkt und intuitive, visuell ansprechende und benutzerfreundliche Projektoberflächen sichergestellt.

200 UI/UX-Verbesserungen vorgestellt

200 UI/UX-Verbesserungen für die GIS-Kartenanwendung vorgestellt und durch erweiterte Softwarefunktionen eine Umsatzsteigerung um das 10-Fache erreicht.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Eine begrenzte Ereignispipeline für eine macOS-App · Nativen macOS-Client in Swift 6 entwickelt · Technische Grundlagen der Plattform festgelegt · Organisationswechsel für Nutzer entwickelt · Eine maritime Plattform mit klaren Systemgrenzen · Full-Stack-GIS-Kartenentwicklung geleitet

{ Softwareentwicklung }

Full-Stack-Produktentwicklung

Ein funktionierendes Produkt, von Anfang bis Ende

Ein Team führt eine Idee von der Skizze bis zu einem Produkt, bei dem Ihre Kunden sich anmelden können, statt sie zwischen drei Dienstleistern weiterzureichen.

Unternehmensbesitzansprüche und Prüfung umgesetzt

Die Bearbeitung von Unternehmensbesitzansprüchen durchgängig umgesetzt: Ein Nutzer beansprucht ein Unternehmen, ein Administrator entscheidet und eine Genehmigung verändert den Berechtigungsgraphen, der die Bearbeitungsrechte festlegt.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Nativen macOS-Client in Swift 6 entwickelt · Datenbankgestützten Lebenslaufgenerator aufgebaut · Hugo-Website mehrsprachig in 3 Sprachen aufgebaut · Stellenmarkt und Seefahrtskarriere entwickelt · Soziale Ebene mit Beiträgen und Gruppen aufgebaut · Verwaltungs-Backend der Plattform entwickelt · Django-Apps für IPTV entwickelt und gewartet · Technische Grundlagen der Plattform festgelegt · Next.js-16-Frontend: SSR, SSG und CSR entwickelt · Eine maritime Plattform mit klaren Systemgrenzen · Ein Kartenprodukt durchgängig verantwortet · 100 Webanwendungen mit HTML/HTML5 entwickelt · 30+ GIS-Projekte von Entwicklung bis Support · Full-Stack-GIS-Kartenentwicklung geleitet · GIS-Kartenprodukt zu 10-fachem Umsatz geführt

{ Softwareentwicklung }

Website-Entwicklung und CMS

Eine Website, die Sie selbst aktualisieren können

Eine schnelle, auffindbare Website, die Ihre Mitarbeitenden selbst bearbeiten können, ohne für jede Textänderung eine Rechnung zu bekommen.

Lesbaren Tor-Onion-Spiegel veröffentlicht

Die Website als Tor-Onion-Spiegel unter einer lesbaren, mit engineer beginnenden Adresse veröffentlicht, den Schlüssel außerhalb des Hosts erzeugt, sodass er weder dieses Repository noch einen Laptop erreichte, und diesen einen von vier Zugängen von der Zählung ausgenommen.

Visuelle Identität aus 2 Zeichnungen entwickelt

Die Symbol- und Favicon-Sätze des Unternehmens aus zwei handgefertigten Zeichnungen aufgebaut, alle abgeleiteten Assets per Skript neu erzeugt und eine Prüfung ergänzt, die vor der Quelle committete Ableitungen zurückweist.

106 von 106 PDF/UA-1-Profilprüfungen bestanden

Die PDF/UA-1-Validierung für neun Dokumente dokumentiert: 106 von 106 Prüfungen des konfigurierten Barrierefreiheitsprofils bestanden, während die Archivvariante eine von 146 Regeln verfehlte.

Sitemap mit einheitlichem Zeitstempel korrigiert

Eine Sitemap korrigiert, in der 172 von 176 URLs denselben Änderungszeitpunkt trugen, indem das Datum nach der Feststellung aus der Git-Historie gelesen wurde, dass der Export jede Datei bei jedem Lauf neu schreibt.

963 Metadatenblöcke durch einen Graphen ersetzt

963 anonyme Blöcke strukturierter Daten, die das Unternehmen 1.671-mal erneut beschrieben, durch einen verknüpften Graphen mit 16 Typen aus stabilen Identifikatoren auf Basis des eigenen Web-Ursprungs ersetzt.

Website nach Gemini und Gopher gespiegelt

Gemini- und Gopher-Ausgaben aus derselben Website-Quelle entwickelt; die Leistung dokumentiert für diesen Stand 777 Dokumente je Format, wobei durch das Hinzufügen der Formate null HTML-Bytes geändert wurden.

495 Seiten aus einem SQLite-Export erzeugt

495 Leistungsseiten in drei Sprachen aus einem schreibgeschützten SQLite-Export erzeugt und die Seitenadresse als eigene Daten festgehalten, damit eine Satzkorrektur nicht länger die Seite verschiebt und Links bricht.

Ihr Code auf Ihrem eigenen Server

Die eigene Git-Forge auf Soft Serve bereitgestellt, standardmäßig privat, ohne Weboberfläche und mit an Loopback gebundenem SSH-Port hinter einem Sprunghost; die vorgeschaltete Startseite als Build-Artefakt der Hauptwebsite statt als manuell gepflegte Kopie umgesetzt.

Webentwicklung mit SEO und Inhalten verbunden

Kunden durch die Verbindung individueller Website-Entwicklung und Gestaltung mit SEO, Inhaltsstrategie und Texterstellung zu einem erfolgreichen Webauftritt verholfen.

40 Websites mit Apache und Nginx administriert

40 Websites auf Ubuntu-Linux-Hostingservern mit Apache und Nginx administriert und hohe Verfügbarkeit und Performance sichergestellt.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Stylesheet von 87 KB auf 31 KB verkleinert · Druckstylesheet mit 637 Zeilen entwickelt · Reduzierte Bewegung wirksam umgesetzt · Barrieren, die das Prüfwerkzeug übersieht · Barrierefreiheitsprüfungen auf 32 Seiten in beiden Farbschemata · Hugo-Website mehrsprachig in 3 Sprachen aufgebaut · Hugo-Website betreut und Produkt-UI verbessert · Django-Apps für IPTV entwickelt und gewartet · Progressive Web App vollständig unterstützt · Next.js-16-Frontend: SSR, SSG und CSR entwickelt · 30 Websites von Photoshop in HTML umgesetzt · 100 Webanwendungen mit HTML/HTML5 entwickelt

{ Softwareentwicklung }

Vorbereitung auf KI-Agenten

Gefunden werden, wenn Kunden eine KI fragen

Eine Website oder App, die die KI-Assistenten Ihrer Kunden finden, lesen und bedienen können — bei jeder Veröffentlichung geprüft statt einmal versprochen.

Entwicklung in 51 schriftlichen Vorhaben geführt

Die Website-Entwicklung als 51 schriftliche Vorhaben in 54 Plandokumenten geführt, jeweils mit offenen Aufgaben, Messungen und verworfenen Entscheidungen — darunter sechs von acht mit dokumentierter Begründung abgelehnte Befunde zur Eignung für KI-Agenten.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Sitemap mit einheitlichem Zeitstempel korrigiert · 963 Metadatenblöcke durch einen Graphen ersetzt · Barrierefreiheitsprüfungen auf 32 Seiten in beiden Farbschemata · Hugo-Website mehrsprachig in 3 Sprachen aufgebaut

{ Softwareentwicklung }

Werkzeuge für KI-Agenten und MCP-Server

KI mit Ihrer vorhandenen Software arbeiten lassen

Ihre bestehende Software wird als Werkzeug für KI-Agenten zugänglich, ohne bereits funktionierende Teile neu zu schreiben.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Verwaltungs-Backend der Plattform entwickelt · Langsame Aufgaben in River-Jobqueue verlagert · Den API-Vertrag aus der Datenbank generiert · API auf Huma v2 migriert, 100 % Übereinstimmung · JSON direkt von Postgres an die API geleitet

{ Softwareentwicklung }

GIS- und Geodatenlösungen

Karten, die Fragen beantworten

Alles, was an einem Ort geschieht — Schiffe, Fahrzeuge, Grundstücke, Sensoren — auf einer Karte, die Sie durchsuchen, vermessen und teilen können.

30 erfolgreiche GIS-Projekte umgesetzt

Die Durchführung von 30 erfolgreichen GIS-Projekten gesteuert und innovative Lösungen im gesamten Fachgebiet unter eigener Leitung umgesetzt.

Prognose- und Investitionsstrategien optimiert

Prognose- und Investitionsstrategien für 11 Stromnetzbetreiber optimiert und die betriebliche Effizienz mit datenbasierten GIS-Lösungen verbessert.

20 GIS-Datenpipelines automatisiert ausgeliefert

Die Bereitstellung von 20 GIS-Datenpipelines und ETL-Prozessen für Anwendungsdaten automatisiert und dadurch Infrastrukturautomatisierung und Berichterstattung vereinfacht.

GIS-SaaS-Deployments mit GitHub automatisiert

Bereitstellung, Datenverarbeitung und Berichtswesen einer GIS-SaaS-Anwendung mit GitHub Actions CI/CD, Python, Bash und SQL automatisiert.

100 PostgreSQL/MS-SQL/BigQuery-Warehouses

100 Data-Warehouse-Datenbanken in PostgreSQL, MS SQL und Google BigQuery mit überwiegend GIS- und Zeitreihendaten konzipiert, aufgebaut und verwaltet sowie Performance und Skalierbarkeit optimiert.

500 Strom- und GIS-Datenberichte veröffentlicht

500 Analyseberichte zu Strom- und GIS-Daten veröffentlicht und durch gründliche Recherche und Fehlerdiagnose präzise Erkenntnisse aus geografischen Big Data und Zeitreihendaten ermöglicht.

1.000 Fehler in geografischen Daten behoben

1.000 Probleme in geografischen Daten und Zeitreihendaten mit GDAL, ArcGIS, PostGIS, Mapbox, QGIS, SQL (PL/pgSQL, Transact-SQL) und Bash behoben und eine hochwertige Big-Data-Verarbeitung sichergestellt.

GIS-Karte mit PostgreSQL 10-mal schneller

Die Performance einer geografischen Kartenanwendung durch einen gezielten Datenbankwechsel von MSSQL zu PostgreSQL einschließlich Datenumschaltung um das 10-Fache verbessert und dabei Verarbeitung und Datensicherheit optimiert.

Geografische Datenpipeline 50-mal schneller

Eine 50-fache Verbesserung der geografischen Datenpipeline bei der Arbeit der Gründerin für Utiligize durch Änderungen an SQL und Datenmodellen in PostgreSQL, MS SQL und Google Cloud BigQuery dokumentiert.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Ein Kartenprodukt durchgängig verantwortet · 30+ GIS-Projekte von Entwicklung bis Support · Full-Stack-GIS-Kartenentwicklung geleitet · GIS-Kartenprodukt zu 10-fachem Umsatz geführt

{ Softwareentwicklung }

UI/UX-Design und Designsysteme

Die Bedienung selbsterklärend machen

Aufbau, Texte und Abläufe werden so gestaltet, dass Menschen ihr Anliegen erledigen können, statt Ihren Support anzurufen.

Fehlermeldungen ohne Schuldzuweisung verfasst

Die Fehlermeldungen der Anwendung anhand eines schriftlichen Tonalitätsstandards neu verfasst, nachdem eine abgewiesene Anmeldung dem Nutzer einen Tippfehler vorwarf, obwohl der Anbieter ein anwendungsspezifisches Passwort verlangte; den Fall mit einem anbieterspezifischen Test abgesichert.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Visuelle Identität aus 2 Zeichnungen entwickelt · macOS-Designsystem mit 48 Tokens aufgebaut · Alle 17 Symbole für Screenreader benannt · Nativen macOS-Client in Swift 6 entwickelt · 106 von 106 PDF/UA-1-Profilprüfungen bestanden · Druckstylesheet mit 637 Zeilen entwickelt · Reduzierte Bewegung wirksam umgesetzt · Fehlende Schicht im hellen Farbschema gefunden · Farbsystem auf 28 Farben begrenzt · Barrieren, die das Prüfwerkzeug übersieht · Barrierefreiheitsprüfungen auf 32 Seiten in beiden Farbschemata · 27 Schriftgrößen durch 6-stufige Skala ersetzt · Webentwicklung mit SEO und Inhalten verbunden · Hugo-Website betreut und Produkt-UI verbessert · UI und UX des Produkts durchgängig gestaltet · iOS-26-'Liquid Glass'-Designsystem entworfen · 30 Websites von Photoshop in HTML umgesetzt · An der Gestaltung von Oberflächen mitgewirkt · 200 UI/UX-Verbesserungen vorgestellt

{ Softwareentwicklung }

Internationalisierung und Lokalisierung

In mehr als einer Sprache funktionieren

Software, die in jedem Ihrer Absatzmärkte natürlich klingt — einschließlich Datumsangaben, Währungen und der Sortierung von Namen.

Sprachübergreifende Zahlenprüfung aufgebaut

Eine sprachübergreifende Inhaltsprüfung entwickelt, die bei fehlenden Zahlen gegenüber dem Englischen und bei sprachfremder Notation fehlschlägt — dabei zwei dänische Beschreibungen ohne eine Kennzahl und sechzehn ukrainische Textstellen mit englischen Anführungszeichen gefunden.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Jeden Text in einen Inhaltsbaum verlagert · Datenbankgestützten Lebenslaufgenerator aufgebaut · Website nach Gemini und Gopher gespiegelt · 495 Seiten aus einem SQLite-Export erzeugt · Barrierefreiheitsprüfungen auf 32 Seiten in beiden Farbschemata · Hugo-Website mehrsprachig in 3 Sprachen aufgebaut · Englisch-dänisches i18n-System aufgebaut

{ Softwareentwicklung }

Entwicklung individueller KI-Agenten

Ein KI-Assistent, der Ihr Fach versteht

Ein Agent wird an Ihre Systeme angebunden, an Fällen Ihrer eigenen Fachleute geprüft und von Ihren Mitarbeitenden beaufsichtigt — ohne Bindung an ein einziges Modell.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Sechs Sammler weisen übersprungene Eingaben aus · Die Prüfwerkzeuge selbst mit Tests geprüft · Commit-Prüfung mit 22 Lintern aufgebaut · Eigene Fehlerüberwachung und Tracing entwickelt · Repository-Prüfengine mit 268 Checks entwickelt

{ Daten und Datenbanken }

Datenbankdesign und Datenmodellierung

Ein sinnvoller Platz für Ihre Daten

Eine Struktur, die zur tatsächlichen Arbeitsweise Ihres Unternehmens passt, damit sich auch spätere Fragen noch beantworten lassen.

Eigene Analysen in PostgreSQL entwickelt

Eigene Produktanalysen in PostgreSQL entwickelt — 47 Funktionen über partitionierten Ereignistabellen mit automatischer Bereinigung — pseudonymisiert mit wechselndem Salt und nur mit Einwilligung des Besuchers.

Zeitlich geordnete Schlüssel für ein wachsendes Datenmodell

Zeitlich geordnete UUID-v7-Identifikatoren (PostgreSQL 18) als Entitätsschlüssel eingeführt, um die Fragmentierung von B-Baum-Indizes zu verringern und Abfragen zu beschleunigen.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Datenbankgestützten Lebenslaufgenerator aufgebaut · Stellenmarkt und Seefahrtskarriere entwickelt · Loyalitätskonto, Ligen und Abzeichen entwickelt · Langsame Aufgaben in River-Jobqueue verlagert · Stripe- und In-App-Kaufberechtigungen entwickelt · 348 Tabellen in 34 Postgres-Schemata modelliert · Kataloggesteuerten Deep-Merge umgesetzt · PostgreSQL-Datenschicht mit Funktionen entworfen · JSON direkt von Postgres an die API geleitet · Eine maritime Plattform mit klaren Systemgrenzen · 100 PostgreSQL/MS-SQL/BigQuery-Warehouses · 6 ETL/ELT-Pipelines entworfen und betrieben

{ Daten und Datenbanken }

Datenbankadministration (DBA)

Die Datenbank in guten Händen

Jemand trägt die Verantwortung für das Fundament des Unternehmens: Es läuft, wird gesichert und verfällt nicht unbemerkt.

1.022 Migrationen mit Schemaabweichungsprüfung

Die Konsistenz des Schemas über 1.022 Migrationen mit einer CI-Prüfung abgesichert, die die Datenbank auf zwei Wegen aufbaut — Neuinstallation sowie Installation mit sämtlichen Migrationen — und bei Unterschieden fehlschlägt.

Sicherungen und Notfallwiederherstellung als Code

Die Aufbewahrung von Datenbanksicherungen als Infrastructure as Code konfiguriert, anschließend die Wiederherstellbarkeit geprüft und das Wiederherstellungsverfahren dokumentiert — mit ausdrücklich benannten Lücken, statt sie erst im Ernstfall entdecken zu lassen.

600 PL/pgSQL-ETL/ELT-Pipelines entwickelt

600 PL/pgSQL-basierte ETL/ELT-Pipelines für effizientere komplexe Datenverarbeitung in mehreren PostgreSQL-Entwicklungs- und Produktivumgebungen entwickelt.

Datenarbeit mit Python und PL/pgSQL automatisiert

Datenverarbeitungsaufgaben mit Shell-Skripten, PL/pgSQL, Python und Transact-SQL automatisiert und Produktivität sowie Effizienz gesteigert.

100 wichtige Sicherungen mit Barman automatisiert

100 geschäftskritische Datensicherungen mit Barman, Google Cloud, Bash und Python automatisiert und die Datenintegrität über die Datenbanken hinweg abgesichert.

PostgreSQL um das 10-Fache beschleunigt

PostgreSQL durch gezielte Indizierung, Partitionierung und Abfrageoptimierung um das 10-Fache beschleunigt und die Datenbankeffizienz für Nutzer-, Mandanten-, Geodaten und Strom-Zeitreihendaten verbessert.

1.000 RBAC-Regeln für sichere Linux-Server

Die Datensicherheit durch 1.000 RBAC-Regeln für Entwickler, Anwendungsinstanzen, PostgreSQL, MS SQL und weitere Linux-Server erhöht, unberechtigte Zugriffe verhindert und die Regeln mit Ansible-Automatisierung dokumentiert.

10 PostgreSQL-/MS-SQL-Server auf Ubuntu

10 PostgreSQL- und MS-SQL-Server auf Ubuntu-Linux-VPS entworfen, bereitgestellt und gewartet und dabei Serverperformance und Zuverlässigkeit sichergestellt.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

100 PostgreSQL/MS-SQL/BigQuery-Warehouses

{ Daten und Datenbanken }

Datenbank-Performanceoptimierung

Langsame Stellen schnell machen

Die Abfragen und fehlenden Indizes hinter einem von Jahr zu Jahr schwerfälligeren System finden und die entscheidenden Probleme beheben.

30 Tage Stromdaten in 180 ms

Varianten zur stündlichen Aggregation des Stromverbrauchs in Python, SQL und Bash + Jq entwickelt; der Python-Benchmark verarbeitete einen 30-Tage-JSONL-Datensatz in 180 ms.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Zeitlich geordnete Schlüssel für ein wachsendes Datenmodell · PostgreSQL um das 10-Fache beschleunigt · 100 PostgreSQL/MS-SQL/BigQuery-Warehouses · GIS-Karte mit PostgreSQL 10-mal schneller · Geografische Datenpipeline 50-mal schneller

{ Daten und Datenbanken }

Datenbankmigration und Modernisierung

Vom alten System wegkommen

Daten aus einer veralteten Lösung in eine unterstützte Umgebung überführen, während der Geschäftsbetrieb weiterläuft.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Jeden Text in einen Inhaltsbaum verlagert · 1.022 Migrationen mit Schemaabweichungsprüfung · GIS-Karte mit PostgreSQL 10-mal schneller

{ Daten und Datenbanken }

Entwicklung von Datenpipelines (ETL/ELT)

Ihre Daten gelangen von selbst von A nach B

Zahlen kommen termingerecht dort an, wo sie gebraucht werden, ohne dass am Monatsende jemand eine Tabelle kopieren muss.

4.000 Stunden durch GitHub-/GitLab-CI/CD gespart

Datenanalyse und Softwareentwicklung durch CI/CD-Praktiken mit GitHub, GitLab, Bash und Python verbessert und 4.000 Stunden eingespart.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

495 Seiten aus einem SQLite-Export erzeugt · Eigene Analysen in PostgreSQL entwickelt · Ein Kartenprodukt durchgängig verantwortet · 600 PL/pgSQL-ETL/ELT-Pipelines entwickelt · Datenarbeit mit Python und PL/pgSQL automatisiert · 30+ GIS-Projekte von Entwicklung bis Support · 20 GIS-Datenpipelines automatisiert ausgeliefert · GIS-SaaS-Deployments mit GitHub automatisiert ·

6 ETL/ELT-Pipelines entworfen und betrieben · 1.000 Fehler in geografischen Daten behoben · Geografische Datenpipeline 50-mal schneller · 30 Tage Stromdaten in 180 ms · Ein Infrastrukturplan für 14 Abteilungen

{ Daten und Datenbanken }

Data-Warehouse-Design

Ein Ort, an dem alle Zahlen zusammenpassen

Vertriebs-, Betriebs- und Finanzdaten werden zusammengeführt, damit zwei Abteilungen nicht länger mit zwei Antworten in dieselbe Besprechung kommen.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

100 PostgreSQL/MS-SQL/BigQuery-Warehouses · 6 ETL/ELT-Pipelines entworfen und betrieben · Geografische Datenpipeline 50-mal schneller

{ Daten und Datenbanken }

Datenanalyse und BI-Dashboards

Sehen, was gerade passiert

Dashboards und Berichte zeigen die wenigen Zahlen, von denen eine Entscheidung tatsächlich abhängt, und aktualisieren sich ohne Aufforderung.

Zeitmanagement- und Berichtssystem entworfen

Ein Zeitmanagement- und Berichtssystem entworfen, das über Jahre ohne wesentliche Änderungen im Produktivbetrieb blieb.

Analysen und regelmäßige Statusberichte geliefert

Analysen und regelmäßige Fortschrittsberichte an den CEO geliefert und technische KPIs sowie Umsetzungsstände in Entscheidungsgrundlagen übersetzt.

Das erste Observability-Dashboard aufgebaut

Das erste Observability-Dashboard des Unternehmens entwickelt und eingeführt, das Systemperformance und Datenvisualisierungen in Echtzeit auf dem großen Bürofernseher anzeigt.

Markenkommunikation anhand von Publikumsreaktionen

Marktforschung und Publikumsrückmeldungen genutzt, um Markenkommunikation und Ansprache von Engineer ApS zu gestalten.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Eigene Analysen in PostgreSQL entwickelt · Ein Datenanalyse-Berichtssystem entwickelt · 4.000 Stunden durch GitHub-/GitLab-CI/CD gespart · Prognose- und Investitionsstrategien optimiert · 500 Strom- und GIS-Datenberichte veröffentlicht · 8 Power-BI-Projekte mit Handbüchern umgesetzt · 30 Tage Stromdaten in 180 ms

{ Daten und Datenbanken }

Data Governance und Datenqualität

Den Zahlen vertrauen können

Regeln und Prüfungen fangen fehlerhafte Daten beim Eingang ab und klären, wer für welche Zahl verantwortlich ist und woher sie stammt.

13 Adressen ohne Tracking gemessen

Dreizehn Adressen ausschließlich anhand des eigenen Webserver-Zugriffsprotokolls gemessen — ohne Cookie, Tracker oder Drittanbieterdienst — das Clientnetz beim Schreiben jeder Zeile maskiert und das Ergebnis dauerhaft mit 21 täglichen Zählwerten und 20 monatlichen Facetten archiviert.

Die Lizenz aller ausgelieferten Komponenten kennen

644 Rust-Crates bei jedem Build auf Lizenzverträglichkeit geprüft und die Wirksamkeit durch das Ändern einer Crate-Lizenz sowie das Verschieben der festgelegten Kernrevision ohne Neuerzeugung nachgewiesen.

Die Prüfwerkzeuge selbst mit Tests geprüft

Tests für die Prüfwerkzeuge selbst geschrieben, nachdem klar war, dass ein nur mit fehlerfreien Eingaben getestetes Werkzeug irgendwann auch nach dem Lesen von nichts Erfolg meldet — einen Tippfehler eingebaut, um die Rechtschreibprüfung zu prüfen, und einen ID-Bereich aus der Datenbank statt einer Testkonstante übernommen.

Zuständigkeit nach Geheimnisleck klar geregelt

Eine Regel zum Zuständigkeitsumfang im Repository festgeschrieben, nachdem eine Umstrukturierung Inventar, Firewallfreigaben und Texte eines anderen Unternehmens hineingetragen hatte; die isolierten Rückstände weiterhin auf Geheimnisse prüfen lassen, statt sie auszuschließen.

Geschäftsanforderungen erhoben und analysiert

Geschäftliche Anforderungen erhoben und analysiert und in umsetzbare Funktionen und User Stories im Einklang mit Data-Governance-Standards übersetzt.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Sprachübergreifende Zahlenprüfung aufgebaut · Jeden Text in einen Inhaltsbaum verlagert · 963 Metadatenblöcke durch einen Graphen ersetzt · Eigene Analysen in PostgreSQL entwickelt · 348 Tabellen in 34 Postgres-Schemata modelliert · Kataloggesteuerten Deep-Merge umgesetzt · PostgreSQL-Datenschicht mit Funktionen entworfen · 600 PL/pgSQL-ETL/ELT-Pipelines entwickelt · 500 Strom- und GIS-Datenberichte veröffentlicht · 1.000 Fehler in geografischen Daten behoben · Ein Infrastrukturplan für 14 Abteilungen

{ Daten und Datenbanken }

Daten für KI nutzbar machen

Daten, mit denen eine KI wirklich arbeiten kann

Ihre Datensätze sind für KI-Agenten erreichbar, jeder Kunde ist nur einmal erfasst, niemand sieht mehr als erlaubt und jede Antwort lässt sich bis zur Quelle zurückverfolgen.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

963 Metadatenblöcke durch einen Graphen ersetzt · 495 Seiten aus einem SQLite-Export erzeugt · 1.022 Migrationen mit Schemaabweichungsprüfung · 348 Tabellen in 34 Postgres-Schemata modelliert · 6 ETL/ELT-Pipelines entworfen und betrieben · Geografische Datenpipeline 50-mal schneller

{ Cloud, Infrastruktur und DevOps }

Cloud-Infrastruktur und Cloud-Migration

Wo die Software läuft

Server werden nach dem tatsächlichen Bedarf dimensioniert und in die Cloud oder aus ihr heraus verlagert — die monatlichen Kosten stehen vorher fest.

Einen zweiten Server aus Code provisioniert

Die Provisionierung eines zweiten Servers bei einem zweiten Cloud-Anbieter automatisiert und die Firewall vor dem Rechner angelegt, damit er von Anfang an geschützt ist; die Firewalls beider Anbieter direkt über deren REST-APIs gesteuert, um eine Drittanbieter-Collection zu vermeiden.

Container mit Podman und Quadlet betrieben

Eine Containerebene mit Podman und Quadlet unter systemd statt Docker bereitgestellt, da Dockers standardmäßige Portfreigaben im Bridge-Netz UFW-Regeln umgehen können; für Bereitstellungen einen unprivilegierten Nutzer mit einem festen Befehl statt root eingerichtet.

Infrastruktur in einem Repository definiert

Die eigene Unternehmensinfrastruktur als 19 Ansible-Playbooks und 34 Rollen mit 12.065 YAML-Zeilen aufgebaut, die einen laufenden Host mit ausschließlich idempotenten Plays in den deklarierten Sollzustand überführen.

Azure-Plattformbereitstellungen verantwortet

Die durchgängige Bereitstellung der Plattform auf Azure verantwortet und Releases über Entwicklungs-, Staging- und Produktivumgebungen hinweg gesteuert.

Azure-Infrastruktur als Code provisioniert

Azure-Infrastruktur mit Bicep als Code provisioniert — Container Apps, PostgreSQL Flexible Server, Front Door/WAF und Netzwerke — für Entwicklungs-, Staging- und Produktivumgebung.

20 Docker-Apps auf Google Cloud und AWS betrieben

20 in Docker containerisierte Anwendungen bereitgestellt und gewartet, Fehler mit Podman und Kubernetes behoben und R-basierte Anwendungen auf Google Cloud und AWS verwaltet.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Öffentliche Dienste auf einem kleinen Server · Die gesamte Kerninfrastruktur neu aufgebaut · Infrastruktur vereinfacht und AWS verlassen

{ Cloud, Infrastruktur und DevOps }

Infrastructure as Code

Server, die Sie aus versionierter Konfiguration neu aufbauen können

Die gesamte Einrichtung wird in Dateien festgehalten, damit sie sich exakt wiederherstellen lässt und nicht nur im Kopf einer einzelnen Person existiert.

Forge-CI-Hook geprobt und 2 Fehler gefunden

Den CI-Hook auf der Forge probeweise ausgeführt und zwei beim bloßen Lesen unerreichbare Fehler gefunden: einen Fallback mit einem nicht auflösbaren Eingabeargument und eine Git-Umgebungsvariable, die der Qualitätsprüfung in den Checkout folgte und 19 Tests scheitern ließ.

Jedes Betriebsgeheimnis je Host getrennt

Die vier betrieblichen Geheimnisse je Host getrennt, nachdem sich gezeigt hatte, dass zwei Hosts mit einer gemeinsamen Totmannschaltung weniger zuverlässig alarmieren als zwei getrennte Schaltungen und eine gemeinsame Sicherungspassphrase die Sicherungen beider Hosts auch in getrennten Repositorys offenlegen kann, wenn jemand die Passphrase erlangt und die verschlüsselten Daten und Schlüsseldateien beider Repositorys lesen kann.

Alle systemd-Units stärker isoliert

Die Angriffsfläche der systemd-Sandboxen für jede von der Plattform installierte Unit verringert — einen Totmannschaltungsdienst von 9,6 UNSAFE auf 1,5, eine internetseitig erreichbare Git-Forge von 8,3 EXPOSED auf 1,5 — und eine Parserprüfung beim Konvergenzlauf ergänzt, nachdem eine falsch geschriebene Direktive in drei Unit-Vorlagen stillschweigend ignoriert worden war.

DNS-Zone mit 20 Einträgen als Code abgeglichen

Eine DNS-Zone mit 20 Einträgen deklarativ über die Cloudflare-API abgeglichen, mit separaten Einstiegspunkten für Audit und BIND-Export, und den zuvor eingerichteten CDN-Proxy aus Datenschutzgründen wieder abgeschaltet.

Sekunden-Preflight für Geheimnisse ergänzt

Ein Preflight-Play ergänzt, das denselben Code wie der Konvergenzlauf mit den lokal beim Betreiber liegenden Geheimnissen in etwa einer Sekunde ausführt, nachdem ein teilweise durchgelaufener Produktivlauf bei Aufgabe neun abbrach, während die Swap-Einstellungen bereits auf dem Produktivhost geschrieben waren.

Ansibles Prüfmodus aussagekräftig gemacht

Den Prüfmodus auf der gesamten Plattform aussagekräftig gemacht, nachdem sechs Sonden anhand eines vom Host nie gelieferten Werts entschieden hatten: Ansibles command-Modul meldet unter --check Erfolg, obwohl es den Befehl vollständig überspringt.

Totmannüberwachung für Hosts entwickelt

Eine Totmannüberwachung entwickelt, die nur bei gesundem Speicher- und Plattenzustand Signale sendet, sodass ein beeinträchtigter Host durch sein Verstummen alarmiert — und sechs Variablennamen mit "free" gefunden, deren Prüfungen korrekt "available" maßen, auf einem Rechner mit 464 MB um eine Größenordnung verschieden.

Sicherungsautomatisierung mit klarem Wiederherstellungsweg

restic-Sicherungsautomatisierung mit Aufbewahrung, Integritätsprüfungen und monatlichem Wiederherstellungstest umgesetzt; Speicheraktivierung und Nachweis einer tatsächlichen Wiederherstellung als getrennte Betriebsschritte dokumentiert.

SSH mit 24 geprüften Direktiven gehärtet

SSH auf 24 geprüfte Direktiven gehärtet und dreistufig validiert — Kandidatendatei, zusammengesetzte Konfiguration und anschließend die Rückmeldung des Daemons selbst — nachdem diese gezeigt hatte, dass der laufende Server zwei der vierundzwanzig Direktiven stillschweigend überschrieb.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Lesbaren Tor-Onion-Spiegel veröffentlicht · Einen zweiten Server aus Code provisioniert · Ihr Code auf Ihrem eigenen Server · Öffentliche Dienste auf einem kleinen Server · Infrastruktur in einem Repository definiert · Sicherungen und Notfallwiederherstellung als Code · Azure-Infrastruktur als Code provisioniert · 1.000 RBAC-Regeln für sichere Linux-Server

{ Cloud, Infrastruktur und DevOps }

DevOps- und CI/CD-Automatisierung

Änderungen ohne Aufregung ausliefern

Geprüft, veröffentlicht und rückgängig zu machen: Änderungen erreichen Ihre Kunden am Dienstagnachmittag statt spät am Freitagabend.

4 Repositorys auf einen Historienstandard geeint

Vier Repositorys auf einen einheitlichen Historienstandard verpflichtet — Conventional Commits, ohne Emojis und Attribution-Trailer, durch einen Commit-Message-Hook erzwungen — ergänzt durch 48 Anweisungsdokumente zur Arbeitsweise.

Testsuite auf 8,9 Sekunden beschleunigt

Die Testsuite von sechs Tests jenseits eines Limits von sechzig Sekunden auf 135 erfolgreiche Tests in 8,9 Sekunden gebracht, indem der Hauptthread profiliert und die zwei Aufrufe entfernt wurden, in denen er bei 3.989 von 4.017 Stichproben festhing.

Abweichende Hooks und Qualitätsprüfung gefunden

Durch den Vergleich beider Listen in einem Test festgestellt, dass Commit-Hooks und Qualitätsprüfung unterschiedliche Prüfungen ausführten, obwohl ein Dokument Gleichheit versprach — die fünf ausschließlich von Hand ausgeführten Prüfungen waren gerade diejenigen für die Lebenslauftexte.

Commit-Prüfung mit 22 Lintern aufgebaut

Eine verbindliche Commit-Prüfung aus 22 einzeiligen Lintern und fünf ausführlicheren Prüfungen aufgebaut, ohne Warnungsstufe und ohne Inline-Unterdrückungen, für HTML, CSS, JavaScript, Python, YAML, Markdown, Shell, Links, Rechtschreibung, Geheimnisse und Typografie.

Kontinuierliche Sicherheitsprüfungen etabliert

Ein kontinuierliches Sicherheitsprogramm aufgebaut — Codeanalyse, DAST, Abhängigkeits- und Schwachstellenprüfungen, SBOM-Erzeugung, Geheimnissuche und per SHA festgelegte Actions — ergänzt durch 21 schriftliche Sicherheitsaudits.

Repository-Prüfengine mit 268 Checks entwickelt

Die Prüfengine des Repositorys entwickelt — 268 registrierte Commit-Prüfungen, 277 Lint-Regeln und 15 eigene ESLint-Regeln — sowie 146 Tests der Prüfungen selbst, damit der Build den Standard durchsetzt und nicht erst das Review.

Teamzusammenarbeit automatisiert

Teamzusammenarbeit, Passwortverwaltung sowie Aufgaben- und Zeitmanagement automatisiert und ein halbautomatisches System zur Projektpräsentation entwickelt, um die Teamproduktivität zu steigern.

Warnungsfreiheit für sechs Sprachen durchgesetzt

Für sechs Sprachen — Go, TypeScript, SQL, Python, Shell und Markdown — einen durch Pre-Commit-Hooks durchgesetzten Qualitätsstandard ohne Warnungen festgelegt.

578 go-task-Automatisierungsziele verfasst

578 go-task-Automatisierungsziele für native, Docker- und HTTPS-Entwicklungsmodi, Linting, Tests, Datenbank und Bereitstellung verfasst.

GitHub-Actions-CI/CD-Pipelines aufgebaut

GitHub-Actions-CI/CD-Pipelines mit einem Distroless-Frontend-Image für den Produktivbetrieb und der Übernahme von Releases über mehrere Umgebungen entwickelt.

CI/CD vereinfacht und 4.000 Stunden gespart

CI/CD-Prozesse verbessert und durch die Automatisierung von Softwareentwicklungspipelines 4.000 Stunden eingespart.

SSL/TLS für 100 Docker-Apps automatisiert

Die Erstellung von SSL/TLS-Zertifikaten für 100 Docker-Anwendungen automatisiert und sichere Verbindungen auf Ubuntu-Linux-Hosts sichergestellt.

Betriebsrisiken mit Grafana/Prometheus gesenkt

Betriebsrisiken durch ein Monitoring-Dashboard mit Grafana und Prometheus verringert und die Systemzuverlässigkeit verbessert.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Die Lizenz aller ausgelieferten Komponenten kennen · Die ungenutzte Hälfte des Kerns erschlossen · Parser zur Prüfung aller FFI-Aufrufe geschrieben · Forge-CI-Hook geprobt und 2 Fehler gefunden · Die Prüfwerkzeuge selbst mit Tests geprüft · Jede Lint-Regel als Fehler aktiviert · Generator mit 981 Tests und 92 % Abdeckung · Qualitätsprüfung mit 10.242 Zeilen selbst geprüft · Sitemap mit einheitlichem Zeitstempel korrigiert · Website nach Gemini und Gopher gespiegelt · Ein Sicherheitsnetz, das keine halbe Stunde mehr kostet ·

Container mit Podman und Quadlet betrieben · Ihr Code auf Ihrem eigenen Server · Sekunden-Preflight für Geheimnisse ergänzt · Ansibles Prüfmodus aussagekräftig gemacht · Infrastruktur in einem Repository definiert · 1.022 Migrationen mit Schemaabweichungsprüfung · Eigene Fehlerüberwachung und Tracing entwickelt · Testsuite für Go, SQL und Browser aufgebaut · Die gesamte Kerninfrastruktur neu aufgebaut · Azure-Plattformbereitstellungen verantwortet · Den API-Vertrag aus der Datenbank generiert · Infrastruktur vereinfacht und AWS verlassen · Datenarbeit mit Python und PL/pgSQL automatisiert · 4.000 Stunden durch GitHub-/GitLab-CI/CD gespart · Abläufe überarbeitet und 8.000 Stunden eingespart · 20 Docker-Apps auf Google Cloud und AWS betrieben · 100 wichtige Sicherungen mit Barman automatisiert · 20 GIS-Datenpipelines automatisiert ausgeliefert · GIS-SaaS-Deployments mit GitHub automatisiert

{ Cloud, Infrastruktur und DevOps }

Containerisierung und Orchestrierung

Software, die überall gleich läuft

So verpackt, dass sie auf dem Laptop, dem Testserver und im Produktivbetrieb gleich funktioniert und bei höherer Auslastung mitwächst.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Container mit Podman und Quadlet betrieben · Alle systemd-Units stärker isoliert · GitHub-Actions-CI/CD-Pipelines aufgebaut · SSL/TLS für 100 Docker-Apps automatisiert · 20 Docker-Apps auf Google Cloud und AWS betrieben

{ Cloud, Infrastruktur und DevOps }

Site Reliability und Monitoring

Bescheid wissen, bevor Ihre Kunden es tun

Monitoring und Alarme melden sich, solange ein Fehler noch klein ist; Kennzahlen zeigen, wie oft das gelingt.

Sechs Sammler weisen übersprungene Eingaben aus

Sechs Berichtssammler um Angaben zu ungelesenen Eingaben ergänzt — ungeöffnete Dateien, nicht installierte Sonden, unbekannte Journalzeilen — und Schwellenwerte anhand von 116 gemessenen Stichproben bestimmt, nachdem eine blinde Erfassung und ein gesunder Host dieselbe Seite ergeben hatten.

7 lesende Host-Berichtsrollen entwickelt

Sieben schreibgeschützte Berichtsrollen entwickelt, die den Zustand eines laufenden Hosts in Markdown abbilden — Fakten, Zugriff, Git, Metriken, Verkehr, Sicherheit und Anbieterinventar — nach der Regel, dass keine Zahl erscheint, die der Lauf nicht gemessen hat.

Angriffssperren bei jedem Lauf nachgewiesen

Den Pfad zur Angriffssperre bei jedem Härtungslauf durchgängig nachgewiesen: eine reservierte Testadresse gesperrt, die entstandene Kernelregel ausgelesen und die Sperre in einem garantierten Aufräumblock aufgehoben, sodass eine ausgefallene Jail den Lauf scheitern lässt, statt einen gesunden Zustand zu melden.

Drei Türschlösser, keines davon abgeschlossen

Drei nie funktionierende SSH-Brute-Force-Schutzmechanismen gefunden und korrigiert: eine Sperr-Jail für Port 22 bei einem auf 1986 lauschenden Daemon, eine durch eine allgemeinere vorherige Regel verdeckte Ratenbegrenzung und eine Sperraktion, deren Programm nie gefunden wurde, sodass nie eine Sperre griff.

IPTV-Signale unterbrechungsfrei übertragen

Die unterbrechungsfreie Übertragung von IPTV-Streaming-Signalen zwischen Anbietern und Kunden sichergestellt und Streaming-Netzwerk sowie IP-Telefonie rund um die Uhr überwacht und gewartet.

24/7-Support für rund 1.000 IPTV/OTT-Server

Rund um die Uhr 24/7-Infrastruktursupport für eine IPTV/OTT-Streaming-Plattform geleistet und rund 1.000 Server sowie kundeneigene Systeme für internationale Kunden in China, den USA und Deutschland administriert.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

13 Adressen ohne Tracking gemessen · Eine begrenzte Ereignispipeline für eine macOS-App · Jedes Betriebsgeheimnis je Host getrennt · Ansibles Prüfmodus aussagekräftig gemacht · Totmannüberwachung für Hosts entwickelt · Sicherungsautomatisierung mit klarem Wiederherstellungsweg · Bei Fehlern sperrenden Missbrauchsschutz gebaut ·

Eigene Fehlerüberwachung und Tracing entwickelt · Langsame Aufgaben in River-Jobqueue verlagert · Infrastruktur entwickelt, die Jahre später läuft · Azure-Plattformbereitstellungen verantwortet · E-Mail mit Failover und Zustellprotokoll gebaut · Ein Kartenprodukt durchgängig verantwortet · 40 Websites mit Apache und Nginx administriert · Betriebsrisiken mit Grafana/Prometheus gesenkt · 10 PostgreSQL-/MS-SQL-Server auf Ubuntu · Das erste Observability-Dashboard aufgebaut

{ Cloud, Infrastruktur und DevOps }

Datensicherung und Notfallwiederherstellung

Nach einem schlechten Tag alles zurückbekommen

Sicherungen, deren Funktion geprüft und nicht vorausgesetzt wird, und ein schriftlicher Plan für den Morgen, an dem ein Server nicht wieder startet.

30 Ubuntu-Linux-VPS-Instanzen verwaltet

30 Ubuntu-Linux-VPS-Instanzen verwaltet, Strategien zur Notfallwiederherstellung umgesetzt und geeignete Netzwerkkonfigurationen sichergestellt.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Jedes Betriebsgeheimnis je Host getrennt ·
Sicherungsautomatisierung mit klarem Wiederherstellungsweg ·
Sicherungen und Notfallwiederherstellung als Code ·
100 wichtige Sicherungen mit Barman automatisiert

{ Cloud, Infrastruktur und DevOps }

Systemadministration

Den Betrieb am Laufen halten

Server, E-Mail, Konten und Updates werden im Alltag betreut, damit die Dinge, auf die alle angewiesen sind, unauffällig funktionieren.

Firewallregeln nach Position geprüft

Firewallregeln nach ihrer Position statt bloß ihrer Existenz geprüft, anhand der nummerierten Regelliste und der aktiven Paketfilterkette, denn eine vorhandene Regel wird nicht zwingend von einem Paket erreicht.

Unternehmensweiten Passwortmanager integriert

Ein unternehmensweites Passwortverwaltungssystem integriert, die Sicherheit erhöht und die Zugriffskontrolle vereinfacht.

100 Bare-Bone-Server administriert

100 Bare-Bone-Server, physische Netzwerke und IP-Telefoniesysteme administriert und eine tragfähige Infrastruktur für das Unternehmenswachstum bereitgestellt.

1.000 WLAN-Router konfiguriert und bereitgestellt

1.000 WLAN-Router konfiguriert und bereitgestellt und die Netzwerkverfügbarkeit und Performance für Kunden verbessert.

Netzwerkinfrastruktur administriert

Die Netzwerkinfrastruktur für über 1.000 Server administriert und eine geeignete Systembereitstellung, Sicherheit und Fehlerdiagnose sichergestellt.

8 WireGuard-/IPsec-VPN-Verbindungen verwaltet

8 WireGuard-VPN- und IPsec-VPN-Verbindungen verwaltet und entstört und eine sichere Kommunikation zwischen Google-Cloud- und Linux-Systemen sichergestellt.

Zugriffsverwaltung mit M365 und SSO geleitet

Durch die Leitung von Vorhaben zur Zugriffsverwaltung mit M365, 1Password, Red Hat SSO und OKTA SSO Sicherheitsverletzungen verhindert.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Lesbaren Tor-Onion-Spiegel veröffentlicht · Sechs Sammler weisen übersprungene Eingaben aus · Forge-CI-Hook geprobt und 2 Fehler gefunden · Ihr Code auf Ihrem eigenen Server · 7 lesende Host-Berichtsrollen entwickelt · Alle systemd-Units stärker isoliert · DNS-Zone mit 20 Einträgen als Code abgeglichen · Totmannüberwachung für Hosts entwickelt · Sicherungsautomatisierung mit klarem Wiederherstellungsweg · Angriffssperren bei jedem Lauf nachgewiesen · SSH mit 24 geprüften Direktiven gehärtet · Öffentliche Dienste auf einem kleinen Server · Infrastruktur in einem Repository definiert ·

Die gesamte Kerninfrastruktur neu aufgebaut · Infrastruktur entwickelt, die Jahre später läuft · IPTV-Signale unterbrechungsfrei übertragen · 24/7-Support für rund 1.000 IPTV/OTT-Server · 40 Websites mit Apache und Nginx administriert · 30 Ubuntu-Linux-VPS-Instanzen verwaltet · 1.000 RBAC-Regeln für sichere Linux-Server · 10 PostgreSQL-/MS-SQL-Server auf Ubuntu

{ Cloud, Infrastruktur und DevOps }

Netzwerke und VPN-Einrichtung

Ihre Standorte und Mitarbeitenden verbinden

WLAN, Firewalls und Fernzugriff werden so eingerichtet, dass Mitarbeitende von überall arbeiten können und nur Berechtigte Zugang erhalten.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:
Einen zweiten Server aus Code provisioniert · DNS-Zone mit 20 Einträgen als Code abgeglichen · Firewallregeln nach Position geprüft · Drei Türschlösser, keines davon abgeschlossen · IPTV-Signale unterbrechungsfrei übertragen · Azure-Infrastruktur als Code provisioniert · 100 Bare-Bone-Server administriert · 1.000 WLAN-Router konfiguriert und bereitgestellt · Netzwerkinfrastruktur administriert · 8 WireGuard-/IPsec-VPN-Verbindungen verwaltet · 30 Ubuntu-Linux-VPS-Instanzen verwaltet

{ Cloud, Infrastruktur und DevOps }

Sicherheits- und Zugriffsverwaltung

Festlegen, wer worauf zugreifen darf

Konten, Passwörter und Berechtigungen werden so geordnet, dass alle genau den beruflich erforderlichen Zugriff haben und ihn am Tag ihres Ausscheidens verlieren.

Teamkommunikation und Zusammenarbeit verbessert

Teamkommunikation und Zusammenarbeit durch die Einführung von Slack, Mattermost, 1Password und Jira verbessert und 8.000 Arbeitsstunden eingespart.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Lesbaren Tor-Onion-Spiegel veröffentlicht · 13 Adressen ohne Tracking gemessen · Die Lizenz aller ausgelieferten Komponenten kennen · Die ungenutzte Hälfte des Kerns erschlossen · Jedes Betriebsgeheimnis je Host getrennt · Zuständigkeit nach Geheimnisleck klar geregelt · Einen zweiten Server aus Code provisioniert · Container mit Podman und Quadlet betrieben · Alle systemd-Units stärker isoliert · DNS-Zone mit 20 Einträgen als Code abgeglichen · Sekunden-Preflight für Geheimnisse ergänzt · Firewallregeln nach Position geprüft · Angriffssperren bei jedem Lauf nachgewiesen · SSH mit 24 geprüften Direktiven gehärtet · Drei Türschlösser, keines davon abgeschlossen · Kontinuierliche Sicherheitsprüfungen etabliert · Unternehmensbesitzansprüche und Prüfung umgesetzt · Bei Fehlern sperrenden Missbrauchsschutz gebaut · Stripe- und In-App-Kaufberechtigungen entwickelt · Unternehmensweiten Passwortmanager integriert · Teamzusammenarbeit automatisiert · Anwendung mit Nonce-basierter CSP gehärtet · Azure-Infrastruktur als Code provisioniert · E-Mail mit Failover und Zustellprotokoll gebaut · Organisationswechsel für Nutzer entwickelt · SSL/TLS für 100 Docker-Apps automatisiert · Netzwerkinfrastruktur administriert · 8 WireGuard-/IPsec-VPN-Verbindungen verwaltet · Zugriffsverwaltung mit M365 und SSO geleitet · 1.000 RBAC-Regeln für sichere Linux-Server

{ Cloud, Infrastruktur und DevOps }

Sicherheit von KI-Agenten

Ihre KI darf nur tun, was sie soll

Der Zugriff jedes Agenten wird erfasst und auf das Nötige begrenzt, jede Handlung außerhalb seiner Änderungsbefugnis protokolliert und jede Schutzmaßnahme geprüft statt bloß vorausgesetzt.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Jedes Betriebsgeheimnis je Host getrennt · Zuständigkeit nach Geheimnisleck klar geregelt · Sekunden-Preflight für Geheimnisse ergänzt · Angriffssperren bei jedem Lauf nachgewiesen · Drei Türschlösser, keines davon abgeschlossen · Kontinuierliche Sicherheitsprüfungen etabliert · Bei Fehlern sperrenden Missbrauchsschutz gebaut · Zugriffsverwaltung mit M365 und SSO geleitet · 1.000 RBAC-Regeln für sichere Linux-Server

{ Beratung und Führung }

Technische Leitung und Beratung

Eine erfahrene Ingenieurin mit am Tisch

Jemand sitzt auf Ihrer Seite des Tisches, wenn eine Lieferanten-, Budget- oder Architekturentscheidung eine zweite fundierte Meinung braucht.

Dokumentation einem eigenen Linter unterstellt

Die Dokumentation einem eigenen Linter unterstellt — mit nur absenkbarem Zeilenbudget je Datei, Inhaltsverzeichnispflicht und Prüfung echter Pfade — der sechs von neun als erledigt markierte Pläne mit offenen Aufgaben und vierzehn in keinem Dokument genannte verpflichtende Commit-Prüfungen fand.

Unternehmensprojekte in der Entwicklung begleitet

Die Umsetzung zahlreicher Unternehmensprojekte begleitet und fachkundige Unterstützung in den Softwareentwicklungsphasen geleistet.

Unternehmen von 4 auf 14 Mitarbeitende ausgebaut

Das Unternehmenswachstum von 4 auf 14 Mitarbeitende mit agilen Methoden und wirksamen Projektmanagementpraktiken geleitet.

50.000 Wörter Softwaredokumentation verfasst

50.000 Wörter umfassender Softwaredokumentation in Markdown auf GitHub, Craft und Confluence verfasst und damit Wissenserhalt und transparente Abläufe sichergestellt.

Geschätzt 4.000 Std. gespart; Team von 2 auf 18

Geschätzt 4.000 Stunden eingespart; zugleich ein von 2 auf 18 Mitglieder wachsendes Team durch Mentoring begleitet sowie Arbeitsabläufe und abteilungsübergreifende Abstimmung verbessert.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

4 Repositorys auf einen Historienstandard geeint · Testsuite auf 8,9 Sekunden beschleunigt · Swift-6-Nebenläufigkeit über C abgesichert · Abweichende Hooks und Qualitätsprüfung gefunden · Die Prüfwerkzeuge selbst mit Tests geprüft · Jede Lint-Regel als Fehler aktiviert · Generator mit 981 Tests und 92 % Abdeckung · Qualitätsprüfung mit 10.242 Zeilen selbst geprüft · Entwicklung in 51 schriftlichen Vorhaben geführt · Ein Sicherheitsnetz, das keine halbe Stunde mehr kostet · Repository-Prüfengine mit 268 Checks entwickelt · Testsuite für Go, SQL und Browser aufgebaut · Analysen und regelmäßige Statusberichte geliefert · Technische Grundlagen der Plattform festgelegt · Warnungsfreiheit für sechs Sprachen durchgesetzt · CI/CD vereinfacht und 4.000 Stunden gespart ·

Abläufe überarbeitet und 8.000 Stunden eingespart · 30+ GIS-Projekte von Entwicklung bis Support · Full-Stack-GIS-Kartenentwicklung geleitet · GIS-Kartenprodukt zu 10-fachem Umsatz geführt · Ein Infrastrukturplan für 14 Abteilungen

{ Beratung und Führung }

Teamentwicklung und Mentoring

Das vorhandene Team weiterentwickeln

Einstellung, Einarbeitung und Coaching sorgen dafür, dass das Wissen bei Ihren eigenen Mitarbeitenden bleibt, statt mit einer Beraterin zu gehen.

Ein Team für IT-Support geleitet

Ein Team für IT-Support, Datenrettung und Hardwarereparaturen für über 1.000 Kunden geleitet und eine hohe Servicequalität sichergestellt.

150 Stunden monatlich in 30 Projekten gespart

Die Projekteffizienz durch optimierte Arbeitsabläufe und Ressourcenplanung verbessert und über 30 Projekte hinweg monatlich 150 Stunden eingespart.

Eine Abteilung für technischen Support aufgebaut

Eine Abteilung für technischen Support aufgebaut, über 10.000 Kunden bei IT-Anliegen und Fehlern unterstützt und Kundenanfragen in einem CRM nachverfolgt.

5.000 Stunden Kartenentwicklung gesteuert

5.000 Stunden Kartenentwicklung geplant und gesteuert und dabei Scrum und Jira für ein wirksames Projektmanagement eingesetzt.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Unternehmensprojekte in der Entwicklung begleitet · Teamkommunikation und Zusammenarbeit verbessert · Unternehmen von 4 auf 14 Mitarbeitende ausgebaut · 30 erfolgreiche GIS-Projekte umgesetzt · Geschätzt 4.000 Std. gespart; Team von 2 auf 18

{ Beratung und Führung }

Projektmanagement (Agile)

Das Projekt zum Abschluss bringen

Jemand verantwortet Umfang, Termine und schwierige Gespräche, damit ein Projekt eine zuständige Person hat statt fünf Meinungen.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

4 Repositorys auf einen Historienstandard geeint · Dokumentation einem eigenen Linter unterstellt · Entwicklung in 51 schriftlichen Vorhaben geführt · Teamzusammenarbeit automatisiert · Zeitmanagement- und Berichtssystem entworfen · Analysen und regelmäßige Statusberichte geliefert · 150 Stunden monatlich in 30 Projekten gespart · Eine Abteilung für technischen Support aufgebaut · Abläufe überarbeitet und 8.000 Stunden eingespart · Unternehmensprojekte in der Entwicklung begleitet · Teamkommunikation und Zusammenarbeit verbessert · Unternehmen von 4 auf 14 Mitarbeitende ausgebaut · 30 erfolgreiche GIS-Projekte umgesetzt · Geschäftsanforderungen erhoben und analysiert · 5.000 Stunden Kartenentwicklung gesteuert · Geschätzt 4.000 Std. gespart; Team von 2 auf 18

{ Beratung und Führung }

Produktstrategie und Anforderungen

Herausfinden, was gebaut werden soll

Geschäftliche Wünsche werden in umsetzbare Entwicklungsaufgaben übersetzt; zugleich wird klar ausgesprochen, was sich nicht zu bauen lohnt.

Eine einheitliche Identität über alle Kanäle

Die Markenidentität von Engineer ApS für Offline- und Onlinematerialien entwickelt und den Auftritt mit dem technischen Portfolio verbunden. Die Gründerin berichtet von einer Steigerung der Markenbekanntheit um den Faktor 200 durch diese Arbeit.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Fehlermeldungen ohne Schuldzuweisung verfasst · Entwicklung in 51 schriftlichen Vorhaben geführt · Commit-Prüfung mit 22 Lintern aufgebaut · Stellenmarkt und Seefahrtskarriere entwickelt · Soziale Ebene mit Beiträgen und Gruppen aufgebaut · Loyalitätskonto, Ligen und Abzeichen entwickelt · Unternehmensbesitzansprüche und Prüfung umgesetzt · Verwaltungs-Backend der Plattform entwickelt · Stripe- und In-App-Kaufberechtigungen entwickelt · Zeitmanagement- und Berichtssystem entworfen · UI und UX des Produkts durchgängig gestaltet · Ein Datenanalyse-Berichtssystem entwickelt · Geschäftsanforderungen erhoben und analysiert · 5.000 Stunden Kartenentwicklung gesteuert · 200 UI/UX-Verbesserungen vorgestellt · Prognose- und Investitionsstrategien optimiert · GIS-Kartenprodukt zu 10-fachem Umsatz geführt · Markenkommunikation anhand von Publikumsreaktionen

{ Beratung und Führung }

Technische Dokumentation

So aufschreiben, dass die nächste Person weiterkommt

Übergaben, Betriebsanleitungen und Handbücher
ermöglichen Ihren Mitarbeitenden, das System zu betreiben,
ohne die ursprünglichen Entwickler anrufen zu müssen.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Sechs Sammler weisen übersprungene Eingaben aus · 4 Repositorys auf einen Historienstandard geeint · Fehlermeldungen ohne Schuldzuweisung verfasst · Parser zur Prüfung aller FFI-Aufrufe geschrieben · Sprachübergreifende Zahlenprüfung aufgebaut · Abweichende Hooks und Qualitätsprüfung gefunden · 106 von 106 PDF/UA-1-Profilprüfungen bestanden · Dokumentation einem eigenen Linter unterstellt · Entwicklung in 51 schriftlichen Vorhaben geführt · Commit-Prüfung mit 22 Lintern aufgebaut · Zuständigkeit nach Geheimnisleck klar geregelt · 7 lesende Host-Berichtsrollen entwickelt · Kontinuierliche Sicherheitsprüfungen etabliert · Repository-Prüfengine mit 268 Checks entwickelt · 578 go-task-Automatisierungsziele verfasst · API auf Huma v2 migriert, 100 % Übereinstimmung · 50.000 Wörter Softwaredokumentation verfasst · 8 Power-BI-Projekte mit Handbüchern umgesetzt · Ein Infrastrukturplan für 14 Abteilungen

{ IT-Support und Hardware }

IT-Support und Helpdesk

Computerhilfe im Alltag

Laptop, Drucker, Anmeldung und Postfach werden schnell und verständlich in Ordnung gebracht, ohne dass sich jemand unkundig fühlen muss.

Laptop-Hardware diagnostiziert und repariert

Laptop-Hardware diagnostiziert und repariert — Displays, Scharniere, Tastaturen, Trackpads, Hauptplatinen und Stromversorgung — sowie Softwareprobleme unter Linux, Windows und Mac behoben.

Daten aus SD-Karten, HDDs, SSDs und RAID gerettet

Daten von unterschiedlichsten Medien gerettet — SD-Karten, HDDs, SSDs, RAID-Verbünden, externen Laufwerken und Mac-Systemen.

4.000 Computerreparaturen gesteuert

4.000 Computerreparaturen gesteuert und eine schnelle, wirksame Behebung von Hardware- und Softwareproblemen sichergestellt.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

24/7-Support für rund 1.000 IPTV/OTT-Server · 1.000 WLAN-Router konfiguriert und bereitgestellt · Ein Team für IT-Support geleitet · Eine Abteilung für technischen Support aufgebaut

{ IT-Support und Hardware }

Hardwarereparatur und Datenrettung

Dateien von einem ausgefallenen Rechner retten

Reparieren, was sich zu reparieren lohnt, und Fotos, Rechnungen oder Zeichnungen von einer nicht mehr funktionierenden Festplatte bergen.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Laptop-Hardware diagnostiziert und repariert · Daten aus SD-Karten, HDDs, SSDs und RAID gerettet · 4.000 Computerreparaturen gesteuert · Ein Team für IT-Support geleitet

{ Marke und Marketing }

Marke, Marketing und SEO

Gefunden werden und passend auftreten

Ein Name, ein Erscheinungsbild und eine Website, die bei der Suche nach Ihrem Angebot auftauchen und überall denselben Eindruck vermitteln.

Diese Leistung ist Teil von Arbeiten, die an anderer Stelle auf dieser Seite aufgeführt sind:

Visuelle Identität aus 2 Zeichnungen entwickelt · macOS-Designsystem mit 48 Tokens aufgebaut · Sitemap mit einheitlichem Zeitstempel korrigiert · Stylesheet von 87 KB auf 31 KB verkleinert · 963 Metadatenblöcke durch einen Graphen ersetzt · Farbsystem auf 28 Farben begrenzt · 495 Seiten aus einem SQLite-Export erzeugt · 27 Schriftgrößen durch 6-stufige Skala ersetzt · Webentwicklung mit SEO und Inhalten verbunden · Markenkommunikation anhand von Publikumsreaktionen · Eine einheitliche Identität über alle Kanäle

Ihre Ausgangslage vergleichen

Wählen Sie eine vergleichbare Arbeit und bestimmen Sie, was in Ihrer eigenen Situation anders ist.

Preise ansehen Kontakt aufnehmen →

Ein passendes Vorgehen besprechen

Bitten Sie uns, aus der passenden Erfahrung ein Vorgehen abzuleiten, das Sie beurteilen können.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Gedruckt von https://engineer.company/de/portfolio/

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.