Engineer Daten

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.

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 →

Nach Kategorie stöbern

SOFTWAREENTWICKLUNG

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

{ 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.

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.

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.

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.

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.

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.

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.

Kataloggesteuerten Deep-Merge umgesetzt

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

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.

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.

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.

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.

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.

{ 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.

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.

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

{ Daten und Datenbanken }

Datenbankmigration und Modernisierung

Vom alten System wegkommen

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

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.

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

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.

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.

Ein Kartenprodukt durchgängig verantwortet

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

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.

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.

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.

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.

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.

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

Eigene Analysen in PostgreSQL entwickelt · 600 PL/pgSQL-ETL/ELT-Pipelines entwickelt · Datenarbeit mit Python und PL/pgSQL automatisiert · 6 ETL/ELT-Pipelines entworfen und betrieben · Geografische Datenpipeline 50-mal schneller · 30 Tage Stromdaten in 180 ms

{ 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.

Ein Datenanalyse-Berichtssystem entwickelt

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

Prognose- und Investitionsstrategien optimiert

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

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.

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.

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

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.

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.

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.

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:

Jeden Text in einen Inhaltsbaum verlagert · 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

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

Mit Sorgfalt und Liebe gestaltet von D.

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

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

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