Engineer Data

Portfolio

Engineering services, with examples of delivered work.

Engineer ApS was registered in 2025. Most of the work shown here was delivered before that by our founder inside client teams; some has been delivered by the company itself, and some since in employment. The same hands and the same standards — we say which is which rather than blurring it.

Find experience worth applying

Use this record to find experience that is relevant to your project.

Explore our services Get in touch →

Read the decisions behind the record

The entries describe delivered work across services. Read a review to see the situation and decisions behind it.

Cases Get in touch →

Browse by category

SOFTWARE ENGINEERING

Backend Engineering · Frontend Engineering · Full-Stack Development · APIs & Integration · Web Development · Testing & QA · Performance Tuning · Python · Internationalization

DATA & DATABASES

Databases · PostgreSQL · SQL · Data Engineering · Data Pipelines (ETL/ELT) · Data Analytics · Data Governance · Data Warehousing · GIS / Geospatial

CLOUD, INFRASTRUCTURE & DEVOPS

Cloud · Containers (Docker/Kubernetes) · DevOps · Automation & CI/CD · Infrastructure · Linux & Servers · System Administration · Networking & VPN · Monitoring & Observability · Reliability & Backups · Security · Platform Architecture · Migrations & Modernization

CONSULTING & LEADERSHIP

Technical Leadership · Team Leadership · Agile & Scrum · Project Management · Product & Requirements · Stakeholder & Reporting · Documentation · Solution Architecture · Brand & Marketing

Everything below **was delivered** — not proposed, not planned.

Each entry names what the work changed and links the services it demonstrates.

What we offer What it costs →

{ Data & Databases }

Database Design & Modeling

Somewhere sensible to keep your data

A structure that matches how the business actually works, so the questions you need answered later are still possible to ask.

Built a database-driven CV generator

Built a database-driven CV, references, portfolio and cover-letter generator in Python — 41 modules, 10,580 lines — rendering six output formats from one 23-table SQLite source assembled by a 19-step idempotent pipeline.

Built the hiring and seafarer career workspaces

Built hiring and seafarer career workspaces; the recorded implementation snapshot contains 151 stored functions across 48 tables for vacancies, applications, certificates, rank progression and sea time.

Built a loyalty wallet, leagues and badges

Built the loyalty and reputation system — 67 functions over a 31-table ledger, with leagues, badges and a redemption shop — taking a row lock on the balance to close the double-spend window.

Built first-party analytics inside PostgreSQL

Built first-party product analytics in PostgreSQL — 47 functions over partitioned event tables that prune themselves — pseudonymised behind a rotating salt and gated on the visitor's consent.

Moved slow work onto a River job queue

Moved slow work off the request path onto a River job queue — 15 worker modules, 8 scheduled tasks and 20 pg_cron jobs — so a request returns while the work behind it carries on.

Built Stripe and in-app purchase entitlements

Built the payments and entitlements layer — Stripe alongside Apple and Google in-app purchase — gating the directory, search and export through an 11-table access model checked on the server.

Modeled 348 tables across 34 Postgres schemas

Modeled the maritime domain into 348 normalized tables across 34 PostgreSQL schemas — professionals, companies, ships, jobs, reviews and the rest — with SMALLINT lookups and UUID v7 keys.

Implemented a catalogue-driven deep-merge

Implemented a catalogue-driven deep-merge for stored JSON preferences, preventing missing-key crashes as the schema evolves.

Time-ordered keys for a growing data model

Adopted UUID v7 time-ordered identifiers (PostgreSQL 18) as entity keys to reduce B-tree index fragmentation and speed up queries.

Designed a function-first PostgreSQL data layer

Designed a PostgreSQL function-first data layer — 1,275 stored functions across 34 schemas — so every read and write goes through a function the database can grant, rather than through a table.

Designed a JSON passthrough from Postgres to API

Designed a JSON passthrough architecture where PostgreSQL functions return complete JSON forwarded verbatim by the Go API, eliminating intermediate unmarshalling and decoupling the frontend from schema changes.

A maritime platform with clear system boundaries

Architected a layered maritime platform separating a Next.js PWA frontend, a Go (Huma/Fiber) API, and a PostgreSQL function layer, keeping all business logic in the database.

Ran 100 PostgreSQL, MS SQL, BigQuery warehouses

Architected, created, and managed 100 PostgreSQL, MS SQL, and Google BigQuery data warehouse databases with primarily GIS and time-series data, optimizing performance and scalability.

Designed and ran 6 ETL/ELT pipelines

Designed, implemented, and administered 6 ETL/ELT pipelines, utilizing Google BigQuery, MSSQL, PostgreSQL, Shell scripting, PL/pgSQL, and Transact-SQL, integrating data for efficient Python API processing.

{ Data & Databases }

Database Administration (DBA)

Looking after the database

Somebody responsible for the thing the whole company rests on: it is up, it is backed up, and it is not quietly rotting.

1,022 migrations behind a schema drift gate

Kept the schema honest across 1,022 migrations with a CI gate that builds the database both ways — a fresh install, and an install plus every migration — and fails when the two disagree.

Implemented backups and disaster recovery as code

Configured database backup retention as infrastructure-as-code, then audited the recovery position and documented the restore procedure — naming the remaining gaps rather than leaving them to be found during an incident.

Engineered 600 PL/pgSQL ETL/ELT pipelines

Engineered 600 PL/pgSQL-based ETL/ELT pipelines to streamline complex data processing workflows across multiple PostgreSQL development and production environments.

Automated data processing in Python and PL/pgSQL

Automated data processing tasks using Shell scripting, PL/pgSQL, Python, and Transact-SQL, increasing productivity and efficiency.

Automated 100 critical data backups using Barman

Automated 100 critical data backups using Barman, Google Cloud, Bash, and Python, ensuring data integrity across databases.

Accelerated PostgreSQL performance by 10x

Accelerated PostgreSQL performance by 10x via strategic indexing, partitioning, and query optimization, enhancing database efficiency for user, tenant, geospatial, and time-series electrical data.

Wrote 1,000 RBAC rules to secure Linux servers

Enhanced data security by implementing 1,000 RBAC rules for developers, application instances, PostgreSQL, MS SQL, and other Linux servers, preventing unauthorized access; documented with Ansible automation.

Ran 10 PostgreSQL and MS SQL servers on Ubuntu

Designed, deployed, and maintained 10 PostgreSQL and MS SQL servers on Ubuntu Linux VPS, ensuring optimal server performance and reliability.

This service is part of work listed elsewhere on this page:

Ran 100 PostgreSQL, MS SQL, BigQuery warehouses

{ Data & Databases }

Database Performance Tuning

Making the slow parts fast

Finding the queries and the missing indexes behind a system that has got heavier every year, and fixing the ones that matter.

Improved GIS map performance 10x on PostgreSQL

Improved geographical map application performance by 10x through strategic database transition from MSSQL to PostgreSQL, including the data cutover, optimizing processing and data security.

Made a geographical data pipeline 50x faster

Recorded a 50x geographical data-pipeline improvement in the founder's Utiligize work through SQL and data-model changes across PostgreSQL, MS SQL and Google Cloud BigQuery.

30 days of electricity data in 180 ms

Engineered hourly electricity-consumption aggregation variants in Python, SQL and Bash + Jq, with the Python benchmark processing one 30-day JSONL dataset in 180ms.

This service is part of work listed elsewhere on this page:

Time-ordered keys for a growing data model · Accelerated PostgreSQL performance by 10x · Ran 100 PostgreSQL, MS SQL, BigQuery warehouses

{ Data & Databases }

Database Migration & Modernization

Getting off the old system

Data taken out of something ageing and into something supported, with the business still running while it happens.

Moved every string into a content tree

Moved every user-facing string out of Python into a content tree of 874 files across three languages, after finding dead translations nobody could see were dead and a check silently grading a third of the achievements.

This service is part of work listed elsewhere on this page:

1,022 migrations behind a schema drift gate · Improved GIS map performance 10x on PostgreSQL

{ Data & Databases }

Data Pipeline Development (ETL/ELT)

Getting your data from A to B on its own

Numbers arriving where they are needed, on a schedule,
without anyone copying a spreadsheet at the end of the month.

Generated 495 pages from a SQLite export

Generated 495 achievement pages across three languages from a read-only SQLite export, with the page address authored as data so that correcting a sentence no longer moved the page and broke the link.

A mapping product, carried end to end

Architected, developed and supported infrastructure, data processing and a map application over 2 years, covering both software delivery and operational maintenance.

Saved 4,000 hours with GitHub and GitLab CI/CD

Streamlined data analysis and software development processes, saving 4,000 hours by introducing GitHub, GitLab, Bash, and Python CI/CD practices.

Led 30+ GIS projects from build to support

Led the development, deployment, and support of over 30 GIS projects, demonstrating expertise in PostgreSQL, Bash, Python, JavaScript, GDAL, ArcGIS, PostGIS, and Mapbox technologies.

Automated delivery of 20 GIS data pipelines

Automated delivery of 20 GIS data pipelines and app data ETL processes, streamlining infrastructure automation and reporting.

Automated GIS SaaS deployment with GitHub Actions

Automated GIS SaaS application deployment, data processing, and reporting system using GitHub Actions CI/CD, Python, Bash, and SQL.

Resolved 1,000 issues in geographical data

Resolved 1,000 issues in geographical data and time-series data, using GDAL, ArcGIS, PostGIS, Mapbox, QGIS, SQL (PL/pgSQL, Transact-SQL), Bash, ensuring high-quality big data processing.

One infrastructure plan for 14 departments

Designed a comprehensive infrastructure framework for a research institute impacting 14 departments, featuring flexible modules, unified data pipelines, and structured support strategies for long-term adoption.

This service is part of work listed elsewhere on this page:

Built first-party analytics inside PostgreSQL · Engineered 600 PL/pgSQL ETL/ELT pipelines · Automated data processing in Python and PL/pgSQL · Designed and ran 6 ETL/ELT pipelines · Made a geographical data pipeline 50x faster · 30 days of electricity data in 180 ms

{ Data & Databases }

Data Warehouse Design

One place where all the numbers agree

Sales, operations and finance figures brought together, so two departments stop bringing two answers to the same meeting.

This service is part of work listed elsewhere on this page:

Ran 100 PostgreSQL, MS SQL, BigQuery warehouses · Designed and ran 6 ETL/ELT pipelines · Made a geographical data pipeline 50x faster

{ Data & Databases }

Data Analytics & BI Dashboards

Seeing what is going on

Dashboards and reports showing the handful of figures a decision actually turns on, refreshed without being asked.

Designed a time-management and reporting system

Designed a time-management and reporting system that remained in production use for years without significant modification.

Delivered analysis and regular progress reports

Delivered analysis and regular progress reports to the CEO, translating engineering KPIs and delivery status into decisions.

Developed a Data Analytics reporting system

Developed a Data Analytics reporting system, increasing quarterly software revenue by 400% through Python-based PDF reports.

Optimized forecasting and investment strategies

Optimized forecasting and investment strategies for 11 electricity grid operators, driving operational efficiency through data-driven GIS solutions.

Released 500 electricity and GIS data reports

Released 500 electricity and GIS data analysis reports, utilizing deep research and troubleshooting to ensure accurate geographic and time series big data insights.

Delivered 8 Power BI projects with manuals

Delivered 8 Power BI projects with comprehensive manuals, integrating Microsoft Power BI tools with NodeJS API and Python FastAPI for effective data analytics and visualization.

Built the company's first observability dashboard

Developed and launched the company's first observability dashboard, providing real-time system performance insights and data visualization on the large office TV.

Brand communication informed by audience response

Used market research and audience feedback to shape Engineer ApS brand communication and outreach.

This service is part of work listed elsewhere on this page:

Built first-party analytics inside PostgreSQL · Saved 4,000 hours with GitHub and GitLab CI/CD · 30 days of electricity data in 180 ms



{ Data & Databases }

Data Governance & Quality

Trusting the numbers

Rules and checks that catch bad data on the way in, and a clear answer to who owns which figure and where it came from.

Measured 13 addresses with no tracking

Measured thirteen addresses from the web server's own access log alone — no cookie, no tracker and no third-party service — masking the client network as each line is written, and folding the result into a permanent archive of 21 daily counts and 20 monthly facets.

Know the licence of everything you ship

Audited 644 Rust crates for licence compatibility on every build, and proved the check fires by rewriting one crate's licence and by moving the pinned core revision without regenerating.

Built a cross-language figure check

Built a cross-language content check that fails when a translation drops a figure the English states, and when a language uses notation it does not use — finding two Danish descriptions missing a metric and sixteen Ukrainian spans quoting in the English style.

Wrote tests for the checkers themselves

Wrote tests for the checkers themselves after establishing that a checker fed only clean input will one day report clean because it read nothing — planting a misspelling to confirm the spell-check finds it, and taking an id range from the database rather than from a number in the test.

Replaced 963 metadata blocks with one graph

Replaced 963 anonymous structured-data blocks that restated the company 1,671 times with one linked graph of 16 types minted from stable origin identifiers.

Wrote a scope rule after a credential leak

Wrote a scope rule into the repository after a restructure carried another company's inventory, firewall allowances and prose into it — and kept the quarantined residue under the secret scanner rather than excluding it.

Gathered and analyzed business requirements

Gathered and analyzed business requirements to translate into actionable features and user stories aligned with data governance standards.

This service is part of work listed elsewhere on this page:

Moved every string into a content tree · Built first-party analytics inside PostgreSQL · Modeled 348 tables across 34 Postgres schemas · Implemented a catalogue-driven deep-merge · Designed a function-first PostgreSQL data layer · Engineered 600 PL/pgSQL ETL/ELT pipelines · Released 500 electricity and GIS data reports · Resolved 1,000 issues in geographical data · One infrastructure plan for 14 departments

{ Data & Databases }

AI Data Readiness

Data an AI can actually use

Your records where AI agents can reach them, one version of each customer, nobody shown more than they should, and every answer traced to its source.

This service is part of work listed elsewhere on this page:

Replaced 963 metadata blocks with one graph · Generated 495 pages from a SQLite export · 1,022 migrations behind a schema drift gate · Modeled 348 tables across 34 Postgres schemas · Designed and ran 6 ETL/ELT pipelines · Made a geographical data pipeline 50x faster

Compare your starting point

Choose a comparable piece of work and identify what differs in your own situation.

See what it costs Get in touch →

Discuss a relevant approach

Ask us to turn the relevant experience into an approach you can evaluate.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

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

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://data.engineer.company/portfolio/