Engineer

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.

Read the work **in its context** — the decisions, delivery and reported outcomes.

See the whole catalogue

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 →

These names **run through the record below** — nothing exotic for your team to inherit.

Python PostgreSQL TypeScript React GitHub Docker

See the whole stack

Browse by category

SOFTWARE ENGINEERING

Backend Engineering
Frontend Engineering
Full-Stack Development
APIs & Integration
Web Development
Design Systems & UI
UX / UI Design
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 · Mentoring & Coaching · 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 →

{ Software Engineering }

Platform & Solution Architecture

Deciding how the whole thing should be built

The shape of a system worked out on paper first, so the parts fit and the cost is known before anyone writes code.

Adopted Swift 6 strict concurrency over C

Adopted Swift 6 complete strict concurrency with no actors, bridging a blocking C event loop to the main actor through one producer, one consumer and one ordering — after establishing that a task per event loses the ordering the interface depends on.

A bounded event pipeline for a macOS app

Bounded event buffers and logging after a recorded macOS incident with 41 MB per second growth and 111 GB of compressed pages on a 36 GB machine; a recorded full test run reduced core log lines from 610,996 to 1,411.

Built a native macOS client in Swift 6

Built a native macOS messaging client in Swift 6 and SwiftUI — 11,141 lines across 53 files — over the C interface of a Rust core linked as a static archive from a pinned revision.

Vendored a QR encoder in 522 lines

Vendored a QR encoder — Reed-Solomon over GF(256), fixed module layout, eight mask patterns — in 522 lines rather than take a third runtime dependency, and verified it by reading the finished matrix back with an independently written decoder.

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.

Public services on a small server

Hosted the company's public services on a single-core 512 MB host — a git forge, a web server for seven domains, Tor, two alternative-protocol servers and intrusion banning — treating 464 MB of usable memory as the design constraint; backup automation is a separate implementation.

Built the entire core infrastructure from scratch

As one of the first hires, designed and built the entire core infrastructure and supporting processes from scratch for a green-energy SaaS startup, laying the foundation for rapid growth.

Built infrastructure that still runs years later

Planned and implemented new infrastructure functionality for internal and external systems, building solutions durable enough to still run years later with minimal change.

Set the platform's founding technical decisions

Set the platform's founding decisions in the weeks after the repository opened in November 2025 — the layering, database-first data access and the zero-warnings bar — and they still hold nine months on.

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.

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.

Migrated the API to Huma v2 at 100% parity

Migrated the HTTP API from Fiber to Huma v2 — 649 paths and 760 operations — reaching and holding 100% parity between the routes the server registers and the OpenAPI description it publishes.

Built organization context switching for users

Designed an organization context-switching system with client localStorage and server-side cookie mirroring, letting users act as managed organizations while enforcing least-privilege authorization.

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.

Infrastructure simplified beyond AWS

Reworked an international client’s infrastructure by removing unused services, adjusting capacity and moving workloads away from AWS. The founder reports a 10x reduction in cloud costs with no loss of productivity.

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.

Overhauled internal processes, saving 8,000 hours

Overhauled internal processes, saving 8,000 hours by improving software architecture, systems, and scheduling efficiency.

Led a GIS map application to 10x revenue growth

Led the software development of a GIS map application, driving revenue growth by 10x and positioning the product as a primary data asset.

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.

{ Software Engineering }

Backend & API Development

The engine behind your app or website

The part users never see: where the data lives, what the rules are, and how other systems are allowed to talk to it.

Reached the unused half of the core

Reached the half of the messaging core the application had never used — backup transfer, disappearing messages, message editing and resending, verified invitations, proxies and encryption policy — driving every test against the real library with no mocks.

Wrote a parser verifying every FFI call

Wrote a parser that reads the real 7,308-line C header and verifies every call site, every enum constant and that every pointer-owning class is final, after a hand-written placeholder header let calls to three removed functions compile, link and crash.

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.

Enabled every linter rule as an error

Selected every rule the Python linter has as an error, working through 1,815 findings to reach zero, with each of the few exemptions carrying a written reason and two of them backed by a checker instead of a comment.

Held the generator to 981 tests and 92% coverage

Held the generator to 981 test cases at a 92% branch-coverage floor with warnings treated as failures, and asserted idempotence by running the whole build pipeline twice from an empty file and requiring the second pass to change nothing.

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 the social layer: posts, feed and groups

Built the platform's social layer — posts, feed, groups, mentions, a follower graph and an occasions digest — on the same function-first data layer as the rest of the product.

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 fail-closed rate limiting and anti-abuse

Built fail-closed abuse controls — 22 Redis-backed rate limiters, Cloudflare Turnstile, request idempotency and an origin lock — so the platform sheds bots and floods instead of trusting its callers.

Built first-party error monitoring and tracing

Built first-party error monitoring and OpenTelemetry tracing rather than buying them — payload sanitising, spike and regression detection, symbolication and a synthetic heartbeat — behind 11 operator views.

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.

Built a test suite across Go, SQL and browser

Built a layered automated test suite — 981 Go tests, 543 frontend and browser specs, 494 SQL behavioural tests — with mutation testing, property-based tests and an accessibility gate.

Built and maintained Django apps for IPTV

Developed and maintained Django web applications for the IPTV platform, shipping new features and improving performance and stability.

Hardened the application with nonce-based CSP

Hardened the application with nonce-based CSP, HSTS, SameSite cookies, least-privilege database roles and server-side entitlement re-checks.

Implemented a catalogue-driven deep-merge

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

Built email with failover and delivery logging

Built email as a platform capability — three providers with failover, delivery webhooks, send and delivery logging, templating and campaigns — behind a startup check that will not boot without one.

Generated the API contract from the database out

Generated the API contract outward from the database — OpenAPI, a 44,076-line typed TypeScript client, 61 mock handlers and the limits the UI enforces — with a guard at every hop that fails on drift.

Developed a Data Analytics reporting system

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

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.

Directed full-stack GIS map development

Directed full-stack GIS map development, overseeing PostgreSQL, Mapbox, ReactJS, and NodeJS to deliver an integrated solution.

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.

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.

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

Adopted Swift 6 strict concurrency over C · Vendored a QR encoder in 522 lines · Built a database-driven CV generator · Set the platform's founding technical decisions · Designed a function-first PostgreSQL data layer · Migrated the API to Huma v2 at 100% parity · Built organization context switching for users · Designed a JSON passthrough from Postgres to API · A maritime platform with clear system boundaries

{ Software Engineering }

Frontend Development

The screens people actually use

Interfaces that load quickly, work on a phone and behave the same in every browser your customers happen to use.

Built a 48-token macOS design system

Built a 48-token design system on the platform's own glass material and wrote the linter that rejects a magic number, a hardcoded font size, or an animation that ignores the reduce-motion preference.

Named all 17 icons for screen readers

Named every icon-only control in the interface for screen readers after finding the send button announced as "arrow up circle, button", and wrote the linter that requires the label within eight lines of the icon.

Linted the 10,242-line quality gate itself

Brought 10,242 lines of quality-gate JavaScript under a formatter and a linter after establishing it was the largest body of code in the repository and the only one nothing read, fixing 13 findings and suppressing none.

Cut the stylesheet bundle from 87 KB to 31 KB

Cut the stylesheet bundle from 87 KB to 31 KB by taking a base64 font out of it, and dropped 756 KB across 22 files that were published on every deploy and referenced by nothing.

Built a 637-line print stylesheet

Built a 637-line print stylesheet against four documented rendering-engine behaviours, including run-in headings that printed white on white for any reader whose browser preferred dark.

Made the reduced-motion setting honest

Made reduced-motion honest after finding eleven selectors still animating under the preference, because a universal transition-none rule loses on specificity to any rule carrying a class.

Found the light theme missing a whole layer

Found that the light theme had been missing a full-page overlay layer since the day it was written, by asserting that both themes paint every layered surface with the same number of layers.

Closed the colour system at 28 colours

Closed the colour system at 28 documented colours with a linter that fails a value painted but undocumented, documented but unpainted, mis-measured, respelled as a literal, or within a perceptual distance of 0.02 of one already there.

The accessibility faults your checker misses

Found and fixed 11 accessibility defects the browser reported as healthy — eight footer links still in tab order behind pointer-events, a scroll timeline clipping the colophon off three pages, and a rule engine running 70 of its 105 rules.

Accessibility checks across 32 pages and both themes

Implemented WCAG 2.2 Level AA checks across 32 representative pages in both colour themes, with a conformance record, one AAA criterion checked and the partial result for criterion 1.4.13 stated explicitly.

Replaced 27 font sizes with a 6-step scale

Replaced 27 rendered font sizes whose nearest neighbours were 0.6% apart with a six-step Major Third scale, and wrote the linter that fails the twenty-eighth.

A safety net that no longer costs half an hour

Cut the site's browser-driven quality gate from 1,636 seconds to 615 by scheduling its checks longest-first through a worker pool bounded to four lanes, after measuring that alphabetical order cost 320 seconds against 224.

Built a multilingual Hugo site in 3 languages

Built a multilingual static site in Hugo across 108 template files including 53 partials, publishing every page in four representations from one content tree in three languages, and again under seven focused subdomains built from that same tree.

Built the operator back-office for the platform

Built the operator back-office — around 40 admin routes and 128 components covering claims, moderation, feature flags, cache and diagnostics — so the platform can be run without database access.

Maintained a Hugo site and improved product UI

Maintained and enhanced the legacy Hugo static-site website while contributing UI/UX improvements to the primary asset-management product.

Built an English/Danish i18n system

Established an English/Danish internationalization system carrying 12,027 messages per locale across 458 namespace files, with linter-enforced vocabulary and a 400-line budget per file.

Designed the product's UI and UX end to end

Designed the product's UI and UX end to end — directory grids, dual card/table views, live requirement validators and breadcrumb navigation.

Designed an iOS 26 'Liquid Glass' design system

Designed an iOS 26 'Liquid Glass' design system and a canonical component inventory enforced by lint rules to prevent UI divergence.

Delivered full Progressive Web App support

Delivered full Progressive Web App support — installable and offline-capable — with Workbox runtime caching via next-pwa.

Built the Next.js 16 frontend: SSR, SSG and CSR

Built the Next.js 16 frontend with deliberate SSR, SSG and CSR strategies, and a reusable prefetched-server-page factory that removes the N+1 fetch cascade from each authenticated page moved onto it.

Designed 30 websites from Photoshop to HTML

Designed 30 websites, delivering unique and flexible solutions by converting Photoshop designs to HTML.

Developed 100 web applications using HTML/HTML5

Developed 100 web applications using HTML/HTML5, CSS/SCSS, Django, WordPress, and Joomla, ensuring diverse online presence.

Contributed to user interface design processes

Contributed to user interface design processes, ensuring intuitive, visually appealing, and user-friendly project interfaces.

Presented 200 UI/UX improvements

Presented 200 UI/UX improvements for the GIS map application, boosting revenue by 10x through enhanced software features.

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

A bounded event pipeline for a macOS app · Built a native macOS client in Swift 6 ·

Set the platform's founding technical decisions · Built organization context switching for users ·

A maritime platform with clear system boundaries · Directed full-stack GIS map development

{ Software Engineering }

Full-Stack Product Development

A working product, end to end

One team taking an idea from a sketch to something your customers can log into, instead of handing it between three suppliers.

Built company ownership claims and adjudication

Built company ownership claims end to end — a user claims a company, an administrator adjudicates, and an approval rewrites the authorization graph that decides who is allowed to edit what.

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

Built a native macOS client in Swift 6 · Built a database-driven CV generator · Built a multilingual Hugo site in 3 languages · Built the hiring and seafarer career workspaces · Built the social layer: posts, feed and groups · Built the operator back-office for the platform · Built and maintained Django apps for IPTV · Set the platform's founding technical decisions · Built the Next.js 16 frontend: SSR, SSG and CSR · A maritime platform with clear system boundaries · A mapping product, carried end to end · Developed 100 web applications using HTML/HTML5 · Led 30+ GIS projects from build to support · Directed full-stack GIS map development · Led a GIS map application to 10x revenue growth

{ Software Engineering }

Website Development & CMS

A website you can update yourself

A site that is quick, findable and editable by your own staff, without a bill for every change of wording.

Published a readable Tor onion mirror

Published the site as a Tor onion mirror on a readable address beginning engineer, with the key generated off the host so it never reached this repository or a laptop, and left it the one front door of four that is never counted.

Built the visual identity from 2 drawings

Built the company's icon and favicon sets from two hand-made drawings, with every derived asset regenerated by script and a check that fails when a derived file was committed before its source.

Passed 106 of 106 PDF/UA-1 profile checks

Recorded PDF/UA-1 validation across nine documents with 106 of 106 configured accessibility-profile checks passing, while the archival variant failed one of 146 rules.

Fixed a sitemap with one shared timestamp

Fixed a sitemap where 172 of 176 URLs shared one modification timestamp, by taking the date from git history after establishing that the export rewrites every file on every run.

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.

Mirrored the site to Gemini and Gopher

Built Gemini and Gopher outputs from the same website source; the achievement records 777 documents in each format at that stage, with zero HTML bytes changed by adding the formats.

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.

Your code on your own server

Deployed the company's own git forge on Soft Serve, private by default with no web panel and its SSH port bound to loopback behind a jump host, and made the landing page in front of it a build artefact of the main site rather than a hand-kept copy.

Combined web development with SEO and content

Drove client web success by combining custom website development and design with SEO, content strategy and copywriting.

Administered 40 websites on Apache and Nginx

Administered 40 websites on Ubuntu Linux hosting servers with Apache and Nginx, ensuring high availability and performance.

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

Cut the stylesheet bundle from 87 KB to 31 KB · Built a 637-line print stylesheet ·

Made the reduced-motion setting honest · The accessibility faults your checker misses ·

Accessibility checks across 32 pages and both themes · Built a multilingual Hugo site in 3 languages ·

Maintained a Hugo site and improved product UI · Built and maintained Django apps for IPTV ·

Delivered full Progressive Web App support · Built the Next.js 16 frontend: SSR, SSG and CSR ·

Designed 30 websites from Photoshop to HTML ·

Developed 100 web applications using HTML/HTML5

{ Software Engineering }

AI Agent Readiness

Showing up when customers ask an AI

A site or an app that the AI assistants your customers use can find, read and act on, checked on every release rather than promised once.

Ran development as 51 written initiatives

Ran the site's development as 51 written initiatives across 54 plan documents, each carrying its open boxes, its measurements and the decisions it declined — including six of eight agent-readiness findings refused with the reason recorded.

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

Fixed a sitemap with one shared timestamp · Replaced 963 metadata blocks with one graph · Accessibility checks across 32 pages and both themes · Built a multilingual Hugo site in 3 languages

{ Software Engineering }

AI Agent Tools & MCP Servers

Letting AI use the software you already have

Your existing software opened up as tools an AI agent can call, without rewriting anything that already works.

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

Built the operator back-office for the platform · Moved slow work onto a River job queue · Generated the API contract from the database out · Migrated the API to Huma v2 at 100% parity · Designed a JSON passthrough from Postgres to API

{ Software Engineering }

GIS & Geospatial Solutions

Maps that answer questions

Anything that happens in a place — vessels, vehicles, plots, sensors — put on a map you can search, measure and share.

Managed 30 successful GIS projects to delivery

Managed the execution of 30 successful GIS projects, demonstrating leadership in delivering innovative solutions across the field.

Optimized forecasting and investment strategies

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

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.

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.

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.

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.

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.

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

A mapping product, carried end to end · Led 30+ GIS projects from build to support · Directed full-stack GIS map development · Led a GIS map application to 10x revenue growth

{ Software Engineering }

UI/UX Design & Design Systems

Making it obvious how to use

Layout, wording and flow arranged so people finish what they came to do instead of ringing your support line.

Rewrote error messages to stop blaming users

Rewrote the application's error messages against a written tone standard after a refused sign-in blamed the user for mistyping when the provider actually required an app-specific password, and covered it with a test that names the provider.

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

Built the visual identity from 2 drawings · Built a 48-token macOS design system · Named all 17 icons for screen readers · Built a native macOS client in Swift 6 · Passed 106 of 106 PDF/UA-1 profile checks · Built a 637-line print stylesheet · Made the reduced-motion setting honest · Found the light theme missing a whole layer · Closed the colour system at 28 colours · The accessibility faults your checker misses · Accessibility checks across 32 pages and both themes · Replaced 27 font sizes with a 6-step scale · Combined web development with SEO and content · Maintained a Hugo site and improved product UI · Designed the product's UI and UX end to end · Designed an iOS 26 'Liquid Glass' design system · Designed 30 websites from Photoshop to HTML · Contributed to user interface design processes · Presented 200 UI/UX improvements

{ Software Engineering }

Internationalization & Localization

Working in more than one language

Software that reads naturally in every market you sell in, including the dates, the currencies and the way names sort.

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.

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

Moved every string into a content tree · Built a database-driven CV generator · Mirrored the site to Gemini and Gopher · Generated 495 pages from a SQLite export · Accessibility checks across 32 pages and both themes · Built a multilingual Hugo site in 3 languages · Built an English/Danish i18n system

{ Software Engineering }

Custom AI Agent Development

An AI assistant that knows your trade

An agent wired to the systems you use, tested on cases your own experts write, overseen by your people and never tied to one model.

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

Made six collectors state what they skipped · Wrote tests for the checkers themselves · Built a 22-linter commit gate · Built first-party error monitoring and tracing · Built a 268-check guard engine for the repo

{ 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 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.

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.

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

Built a database-driven CV generator · Built the hiring and seafarer career workspaces · Built a loyalty wallet, leagues and badges · Moved slow work onto a River job queue · Built Stripe and in-app purchase entitlements · Modeled 348 tables across 34 Postgres schemas · Implemented a catalogue-driven deep-merge · Designed a function-first PostgreSQL data layer · Designed a JSON passthrough from Postgres to API · A maritime platform with clear system boundaries · Ran 100 PostgreSQL, MS SQL, BigQuery warehouses · Designed and ran 6 ETL/ELT pipelines

{ 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.

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 · Improved GIS map performance 10x on PostgreSQL · Made a geographical data pipeline 50x faster

{ 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.

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

Moved every string into a content tree · 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.

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.

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

Generated 495 pages from a SQLite export · Built first-party analytics inside PostgreSQL · A mapping product, carried end to end · Engineered 600 PL/pgSQL ETL/ELT pipelines · Automated data processing in Python and PL/pgSQL · Led 30+ GIS projects from build to support · Automated delivery of 20 GIS data pipelines · Automated GIS SaaS deployment with GitHub Actions ·

Designed and ran 6 ETL/ELT pipelines · Resolved 1,000 issues in geographical data · Made a geographical data pipeline 50x faster · 30 days of electricity data in 180 ms · One infrastructure plan for 14 departments

{ 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.

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 · Developed a Data Analytics reporting system · Saved 4,000 hours with GitHub and GitLab CI/CD · Optimized forecasting and investment strategies · Released 500 electricity and GIS data reports · Delivered 8 Power BI projects with manuals · 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.

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.

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:

Built a cross-language figure check · Moved every string into a content tree · Replaced 963 metadata blocks with one graph · 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

{ Cloud, Infrastructure & DevOps }

Cloud Infrastructure & Migration

Where the software runs

Servers sized for what you actually use, moved into the cloud or back out of it, with the monthly bill worked out first.

Provisioned a second server from code

Automated the provisioning of a second server on a second cloud provider, creating the firewall before the machine so it is born behind one, with both providers' firewalls written directly against their REST APIs to avoid a third-party collection.

Ran containers on Podman and Quadlet

Deployed a container plane on Podman and Quadlet under systemd rather than Docker, because Docker's default bridge-port publishing can bypass UFW rules — and gave deploys an unprivileged user with one fixed command instead of root.

Infrastructure defined in a repository

Built the company's own infrastructure as 19 Ansible playbooks and 34 roles across 12,065 lines of YAML, converging a live host to a declared state with every play idempotent.

Owned end-to-end platform deployments to Azure

Owned end-to-end deployments of the platform to Azure, managing releases across development, staging and production environments.

Provisioned Azure infrastructure as code

Provisioned Azure infrastructure as code with Bicep — Container Apps, PostgreSQL Flexible Server, Front Door/WAF and networking — across the development, staging and production environments.

Ran 20 Docker apps on Google Cloud and AWS

Deployed and maintained 20 Docker containerized applications, troubleshooting with Podman and Kubernetes, and managing R-based apps on Google Cloud and AWS.

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

Public services on a small server · Built the entire core infrastructure from scratch · Infrastructure simplified beyond AWS

{ Cloud, Infrastructure & DevOps }

Infrastructure as Code

Servers you can rebuild from versioned configuration

The whole setup written down as files, so it can be recreated exactly and never lives only in one person's head.

Rehearsed the forge CI hook, found 2 defects

Rehearsed the forge-side CI hook and found two defects unreachable by reading the file: a fallback that put an unresolvable argument on the hook's input, and git's own environment variable following the gate into the checkout and turning 19 tests red.

Split every operational secret per host

Split the four per-host operational secrets after establishing that two hosts sharing one dead-man's switch alert less than two switches, not more, and that a shared backup passphrase can expose both hosts' backups, even in separate repositories, if someone obtains it and can read both repositories' encrypted data and key files.

Cut systemd sandbox exposure on every unit

Cut systemd sandbox exposure across every unit this platform installs — a dead-man's-switch service from 9.6 UNSAFE to 1.5, an internet-facing git forge from 8.3 EXPOSED to 1.5 — and added a converge-time parser check after finding a misspelled directive silently ignored in three unit templates.

Reconciled a 20-record DNS zone as code

Reconciled a 20-record DNS zone declaratively against the Cloudflare API with separate audit and BIND-export entry points, and turned the CDN proxy back off on privacy grounds after building it.

Added a one-second preflight for secrets

Added a preflight play that runs the same code as the converge against operator-local secrets in about a second, after a half-applied production run died on its ninth task with the swap settings already written to the live host.

Made Ansible check mode tell the truth

Made check mode tell the truth across the whole platform after finding six probes deciding on a value the host never gave, because Ansible's command module reports success under --check while skipping the command entirely.

Built dead-man's-switch host monitoring

Built dead-man's-switch monitoring that pings only while memory and disk are healthy, so a degraded host raises an alert by going silent — and caught six variable names saying "free" where the check correctly measured "available", an order of magnitude apart on a 464 MB box.

Backup automation with an explicit restore method

Implemented restic backup automation with retention, integrity checks and a monthly restore-test workflow; documented storage activation and live-restore verification as separate operational steps.

Hardened SSH to 24 validated directives

Hardened SSH to 24 asserted directives with three-stage validation — the candidate file, the assembled config, then the daemon's own read-back — after the read-back caught the running server silently overriding two of the twenty-four.

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

Published a readable Tor onion mirror · Provisioned a second server from code ·

Your code on your own server · Public services on a small server ·

Infrastructure defined in a repository · Implemented backups and disaster recovery as code ·

Provisioned Azure infrastructure as code · Wrote 1,000 RBAC rules to secure Linux servers

{ Cloud, Infrastructure & DevOps }

DevOps & CI/CD Automation

Shipping changes without drama

Tested, released and reversible, so changes reach customers on a Tuesday afternoon rather than late on a Friday night.

Held 4 repositories to one history standard

Held four repositories to one history standard — conventional, emoji-free, no attribution trailers, enforced by a commit-message hook — alongside 48 instruction documents that govern how the work is done.

Took the test suite to 8.9 seconds

Took the test suite from six tests over a sixty-second limit to 135 passing in 8.9 seconds by profiling the main thread and removing the two calls it sat inside for 3,989 samples out of 4,017.

Found the hooks and the gate disagreeing

Found the commit hooks and the quality gate running different checks while a document promised they were the same, by comparing the two lists in a test — the five that only ever ran by hand were the ones reading the CV prose.

Built a 22-linter commit gate

Built a commit gate of 22 one-line linters plus five that earn a paragraph, with no warning tier and no inline suppressions permitted, covering HTML, CSS, JavaScript, Python, YAML, Markdown, shell, links, spelling, secrets and typography.

Ran a continuous security scanning programme

Established a continuous security programme — code scanning, DAST, dependency and vulnerability checks, SBOM generation, secret scanning and SHA-pinned actions — alongside 21 written security audits.

Built a 268-check guard engine for the repo

Built the repository's guard engine — 268 registered commit checks, 277 lint rules and 15 custom ESLint rules — plus 146 tests of the guards themselves, so the build holds the standard, not review.

Automated team collaboration

Automated team collaboration, password management, task and time management, and built a semi-automatic project-showcase system, raising team productivity.

Set a zero-warnings bar across six languages

Set a zero-warnings quality bar across six languages — Go, TypeScript, SQL, Python, Shell and Markdown — enforced by pre-commit hooks.

Authored 578 go-task automation targets

Authored 578 go-task automation targets spanning native, Docker and HTTPS dev modes, linting, testing, database and deployment.

Built GitHub Actions CI/CD pipelines

Built GitHub Actions CI/CD pipelines with a distroless production frontend image and multi-environment promotion.

Streamlined CI/CD processes, saving 4,000 hours

Streamlined CI/CD processes, saving 4,000 hours by introducing automation in software development pipelines.

Automated SSL/TLS certs for 100 Docker apps

Automated SSL/TLS certificate creation for 100 Docker applications, ensuring secure connections across Ubuntu Linux hosts.

Cut operational risk with Grafana and Prometheus

Mitigated operational risks by implementing a monitoring dashboard using Grafana and Prometheus, improving system reliability.

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

Know the licence of everything you ship · Reached the unused half of the core · Wrote a parser verifying every FFI call · Rehearsed the forge CI hook, found 2 defects · Wrote tests for the checkers themselves · Enabled every linter rule as an error · Held the generator to 981 tests and 92% coverage · Linted the 10,242-line quality gate itself · Fixed a sitemap with one shared timestamp · Mirrored the site to Gemini and Gopher · A safety net that no longer costs half an hour · Ran containers on Podman and Quadlet · Your code on your own server · Added a one-second preflight for secrets · Made Ansible check mode tell the truth · Infrastructure defined in a repository · 1,022 migrations behind a schema drift gate · Built first-party error monitoring and tracing · Built a test suite across Go, SQL and browser · Built the entire core infrastructure from scratch · Owned end-to-end platform deployments to Azure · Generated the API contract from the database out · Infrastructure simplified beyond AWS · Automated data processing in Python and PL/pgSQL · Saved 4,000 hours with GitHub and GitLab CI/CD · Overhauled internal processes, saving 8,000 hours · Ran 20 Docker apps on Google Cloud and AWS · Automated 100 critical data backups using Barman · Automated delivery of 20 GIS data pipelines · Automated GIS SaaS deployment with GitHub Actions

{ Cloud, Infrastructure & DevOps }

Containerization & Orchestration

Software that runs the same everywhere

Packaged to behave identically on a laptop, on a test server and in production, and to scale up when a busy week arrives.

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

Ran containers on Podman and Quadlet · Cut systemd sandbox exposure on every unit · Built GitHub Actions CI/CD pipelines · Automated SSL/TLS certs for 100 Docker apps · Ran 20 Docker apps on Google Cloud and AWS

{ Cloud, Infrastructure & DevOps }

Site Reliability & Monitoring

Knowing before your customers do

Monitoring and alerts that speak up while a fault is still small, and the figures that show how often it stays that way.

Made six collectors state what they skipped

Made six reporting collectors state the part of their input they never read — files that stayed shut, probes not installed, journal lines unrecognised — and fitted thresholds to 116 measured samples, after establishing that a blind collection and a healthy host render the same page.

Built 7 read-only host reporting roles

Built seven read-only reporting roles that render a live host to Markdown — facts, access, git, metrics, traffic, security and provider inventory — under a rule that no number is printed the run did not measure.

Proved the intrusion ban works every run

Proved the intrusion-banning path end to end on every hardening run by banning a reserved test address, reading the resulting kernel rule and unbanning in a guaranteed cleanup block, so a jail that stops working fails a run instead of reporting healthy.

Three locks on the door, none of them locked

Found and closed three SSH brute-force protections that had never worked: a ban jail watching port 22 while the daemon listened on 1986, a rate limit shadowed by a broader rule above it, and a ban action whose binary never resolved, so no ban had ever applied.

Kept IPTV streaming signals running non-stop

Ensured uninterrupted delivery of IPTV streaming signals between suppliers and clients, monitoring and maintaining the streaming network and IP telephony around the clock.

Ran 24/7 support for ~1,000 IPTV/OTT servers

Provided round-the-clock 24/7 infrastructure support for an IPTV/OTT streaming platform, administering ~1,000 servers plus client-owned systems for global customers in China, the US and Germany.

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

Measured 13 addresses with no tracking · A bounded event pipeline for a macOS app · Split every operational secret per host · Made Ansible check mode tell the truth · Built dead-man's-switch host monitoring · Backup automation with an explicit restore method · Built fail-closed rate limiting and anti-abuse · Built first-party error monitoring and tracing · Moved slow work onto a River job queue · Built infrastructure that still runs years later · Owned end-to-end platform deployments to Azure · Built email with failover and delivery logging · A mapping product, carried end to end · Administered 40 websites on Apache and Nginx · Cut operational risk with Grafana and Prometheus · Ran 10 PostgreSQL and MS SQL servers on Ubuntu · Built the company's first observability dashboard

{ Cloud, Infrastructure & DevOps }

Backup & Disaster Recovery

Getting everything back after a bad day

Backups that are tested rather than assumed, and a written plan for the morning a server does not come back on.

Managed 30 Ubuntu Linux VPS instances

Managed 30 Ubuntu Linux VPS instances, implementing disaster recovery strategies and ensuring optimal network configurations.

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

Split every operational secret per host · Backup automation with an explicit restore method · Implemented backups and disaster recovery as code · Automated 100 critical data backups using Barman

{ Cloud, Infrastructure & DevOps }

System Administration

Keeping the lights on

Day-to-day care of servers, mail, accounts and updates, so the things everybody relies on stay boring.

Verified firewall rules by position

Verified firewall rules by position rather than presence, reading the numbered rule list and the live packet-filter chain, because a rule that exists is not a rule any packet reaches.

Integrated a company-wide password manager

Integrated a company-wide password-management system, strengthening security and streamlining access control.

Administered 100 Bare Bone servers

Administered 100 Bare Bone servers, physical networks, and IP telephony systems, ensuring robust infrastructure for company growth.

Configured and deployed 1,000 Wi-Fi routers

Configured and deployed 1,000 Wi-Fi routers, improving network accessibility and performance for clients.

Administered network infrastructure

Administered network infrastructure for over 1,000 servers, ensuring optimal system deployment, security, and troubleshooting.

Managed 8 WireGuard and IPsec VPN connections

Managed and troubleshot 8 WireGuard VPN and IPsec VPN connections, ensuring secure communication across Google Cloud and Linux systems.

Led access management across M365 and SSO

Prevented security breaches by leading access management initiatives, utilizing M365, 1Password, Red Hat SSO, and OKTA SSO.

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

Published a readable Tor onion mirror · Made six collectors state what they skipped · Rehearsed the forge CI hook, found 2 defects · Your code on your own server · Built 7 read-only host reporting roles · Cut systemd sandbox exposure on every unit ·

Reconciled a 20-record DNS zone as code · Built dead-man's-switch host monitoring · Backup automation with an explicit restore method · Proved the intrusion ban works every run · Hardened SSH to 24 validated directives · Public services on a small server · Infrastructure defined in a repository · Built the entire core infrastructure from scratch · Built infrastructure that still runs years later · Kept IPTV streaming signals running non-stop · Ran 24/7 support for ~1,000 IPTV/OTT servers · Administered 40 websites on Apache and Nginx · Managed 30 Ubuntu Linux VPS instances · Wrote 1,000 RBAC rules to secure Linux servers · Ran 10 PostgreSQL and MS SQL servers on Ubuntu

{ Cloud, Infrastructure & DevOps }

Networking & VPN Setup

Connecting your offices and your people

Wi-Fi, firewalls and remote access set up so staff can work from anywhere and only the right people get in.

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

Provisioned a second server from code · Reconciled a 20-record DNS zone as code · Verified firewall rules by position · Three locks on the door, none of them locked · Kept IPTV streaming signals running non-stop · Provisioned Azure infrastructure as code · Administered 100 Bare Bone servers · Configured and deployed 1,000 Wi-Fi routers · Administered network infrastructure · Managed 8 WireGuard and IPsec VPN connections · Managed 30 Ubuntu Linux VPS instances

{ Cloud, Infrastructure & DevOps }

Security & Access Management

Deciding who can reach what

Accounts, passwords and permissions arranged so people hold exactly the access their job needs, and leavers lose it that day.

Improved team communication and collaboration

Improved team communication and collaboration by implementing Slack, Mattermost, 1Password, and Jira, saving 8,000 hours of labor.

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

Published a readable Tor onion mirror · Measured 13 addresses with no tracking · Know the licence of everything you ship · Reached the unused half of the core · Split every operational secret per host · Wrote a scope rule after a credential leak · Provisioned a second server from code · Ran containers on Podman and Quadlet · Cut systemd sandbox exposure on every unit · Reconciled a 20-record DNS zone as code · Added a one-second preflight for secrets · Verified firewall rules by position · Proved the intrusion ban works every run · Hardened SSH to 24 validated directives · Three locks on the door, none of them locked · Ran a continuous security scanning programme · Built company ownership claims and adjudication · Built fail-closed rate limiting and anti-abuse · Built Stripe and in-app purchase entitlements · Integrated a company-wide password manager · Automated team collaboration · Hardened the application with nonce-based CSP · Provisioned Azure infrastructure as code · Built email with failover and delivery logging · Built organization context switching for users · Automated SSL/TLS certs for 100 Docker apps · Administered network infrastructure · Managed 8 WireGuard and IPsec VPN connections · Led access management across M365 and SSO · Wrote 1,000 RBAC rules to secure Linux servers

{ Cloud, Infrastructure & DevOps }

AI Agent Security

Making sure your AI only touches what it should

Every agent's access mapped and held to what it needs, every action logged where the agent cannot alter it, and every safeguard tested rather than trusted.

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

Split every operational secret per host · Wrote a scope rule after a credential leak · Added a one-second preflight for secrets · Proved the intrusion ban works every run · Three locks on the door, none of them locked · Ran a continuous security scanning programme · Built fail-closed rate limiting and anti-abuse · Led access management across M365 and SSO · Wrote 1,000 RBAC rules to secure Linux servers

{ Consulting & Leadership }

Technical Leadership & Consulting

A senior engineer in the room

Somebody on your side of the table when a supplier, a budget or an architecture decision needs a second informed opinion.

Put the documentation under its own linter

Put the documentation under its own linter — a per-file line budget that only ratchets down, an index requirement and a real-path check — which found six of nine plans marked done while carrying unchecked boxes, and fourteen commit-gating tasks named in no document.

Guided company projects through development

Guided the execution of numerous company projects, offering expert support in the software development phases.

Led company growth from 4 to 14 employees

Led company growth from 4 to 14 employees by employing Agile methodologies and effective project management practices.

Wrote 50,000 words of software documentation

Wrote 50,000 words of comprehensive software documentation utilizing Markdown in GitHub, Craft, and Confluence, thereby ensuring the retention of knowledge and the transparency of the process.

Estimated 4,000 hours saved, team from 2 to 18

Estimated 4,000 hours saved while mentoring a team that grew from 2 to 18 members, improving workflows and cross-departmental coordination.

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

Held 4 repositories to one history standard · Took the test suite to 8.9 seconds · Adopted Swift 6 strict concurrency over C · Found the hooks and the gate disagreeing · Wrote tests for the checkers themselves · Enabled every linter rule as an error · Held the generator to 981 tests and 92% coverage · Linted the 10,242-line quality gate itself · Ran development as 51 written initiatives · A safety net that no longer costs half an hour · Built a 268-check guard engine for the repo · Built a test suite across Go, SQL and browser · Delivered analysis and regular progress reports · Set the platform's founding technical decisions · Set a zero-warnings bar across six languages · Streamlined CI/CD processes, saving 4,000 hours · Overhauled internal processes, saving 8,000 hours · Led 30+ GIS projects from build to support · Directed full-stack GIS map development · Led a GIS map application to 10x revenue growth · One infrastructure plan for 14 departments

{ Consulting & Leadership }

Team Building & Mentoring

Growing the team you already have

Hiring, onboarding and coaching, so the knowledge settles with your own people rather than leaving with a consultant.

Managed a team delivering IT support

Managed a team delivering IT support, data recovery, and hardware repair services to over 1,000 clients, ensuring high-quality service.

Saved 150 hours a month across 30 projects

Enhanced project efficiency, saving 150 hours per month across 30 projects by optimizing workflows and resource management.

Established a Technical Support department

Established a Technical Support department, servicing over 10,000 clients with IT support and troubleshooting solutions, tracking client requests in a CRM.

Managed 5,000 hours of map development

Designed and managed 5,000 hours of map development, using Scrum and Jira for effective project management.

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

Guided company projects through development · Improved team communication and collaboration · Led company growth from 4 to 14 employees · Managed 30 successful GIS projects to delivery · Estimated 4,000 hours saved, team from 2 to 18



{ Consulting & Leadership }

Project Management (Agile)

Getting it delivered

Somebody accountable for scope, dates and the awkward conversations, so a project has one owner instead of five opinions.

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

Held 4 repositories to one history standard · Put the documentation under its own linter · Ran development as 51 written initiatives · Automated team collaboration · Designed a time-management and reporting system · Delivered analysis and regular progress reports · Saved 150 hours a month across 30 projects · Established a Technical Support department · Overhauled internal processes, saving 8,000 hours · Guided company projects through development · Improved team communication and collaboration · Led company growth from 4 to 14 employees · Managed 30 successful GIS projects to delivery · Gathered and analyzed business requirements · Managed 5,000 hours of map development · Estimated 4,000 hours saved, team from 2 to 18

{ Consulting & Leadership }

Product Strategy & Requirements

Working out what to build

Turning what the business wants into something a developer can build, and saying out loud what is not worth building.

A consistent identity across channels

Developed Engineer ApS brand identity across offline and online material, connecting the presentation with its engineering portfolio. The founder reports a 200x increase in brand recognition from this work.

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

Rewrote error messages to stop blaming users · Ran development as 51 written initiatives · Built a 22-linter commit gate · Built the hiring and seafarer career workspaces · Built the social layer: posts, feed and groups · Built a loyalty wallet, leagues and badges · Built company ownership claims and adjudication · Built the operator back-office for the platform · Built Stripe and in-app purchase entitlements · Designed a time-management and reporting system · Designed the product's UI and UX end to end · Developed a Data Analytics reporting system · Gathered and analyzed business requirements · Managed 5,000 hours of map development · Presented 200 UI/UX improvements · Optimized forecasting and investment strategies · Led a GIS map application to 10x revenue growth · Brand communication informed by audience response

{ Consulting & Leadership }

Technical Documentation

Writing it down so the next person can follow

Handovers, runbooks and manuals that let your own staff operate the system without ringing whoever built it.

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

Made six collectors state what they skipped · Held 4 repositories to one history standard · Rewrote error messages to stop blaming users · Wrote a parser verifying every FFI call · Built a cross-language figure check · Found the hooks and the gate disagreeing · Passed 106 of 106 PDF/UA-1 profile checks · Put the documentation under its own linter · Ran development as 51 written initiatives · Built a 22-linter commit gate · Wrote a scope rule after a credential leak · Built 7 read-only host reporting roles · Ran a continuous security scanning programme · Built a 268-check guard engine for the repo · Authored 578 go-task automation targets · Migrated the API to Huma v2 at 100% parity · Wrote 50,000 words of software documentation · Delivered 8 Power BI projects with manuals · One infrastructure plan for 14 departments

{ IT Support & Hardware }

IT Support & Helpdesk

Everyday computer help

The laptop, the printer, the login and the mailbox, sorted quickly and in plain language, without anyone feeling foolish.

Diagnosed and repaired laptop hardware

Diagnosed and repaired laptop hardware — screens, hinges, keyboards, trackpads, motherboards and power — and resolved software issues across Linux, Windows and Mac.

Recovered data from SD cards, HDDs, SSDs and RAID

Performed data recovery across a wide range of media — SD cards, HDDs, SSDs, RAID arrays, external drives and Mac systems.

Managed 4,000 computer repairs

Managed 4,000 computer repairs, ensuring rapid and effective resolution of hardware and software issues.

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

Ran 24/7 support for ~1,000 IPTV/OTT servers · Configured and deployed 1,000 Wi-Fi routers · Managed a team delivering IT support · Established a Technical Support department

{ IT Support & Hardware }

Hardware Repair & Data Recovery

Rescuing files from a dead machine

Repairing what is worth repairing, and getting the photographs, invoices or drawings off a disk that has stopped working.

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

Diagnosed and repaired laptop hardware · Recovered data from SD cards, HDDs, SSDs and RAID · Managed 4,000 computer repairs · Managed a team delivering IT support

{ Brand & Marketing }

Brand, Marketing & SEO

Being found and looking the part

A name, a look and a website that turn up when somebody searches for what you sell, and read the same everywhere you appear.

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

Built the visual identity from 2 drawings · Built a 48-token macOS design system · Fixed a sitemap with one shared timestamp · Cut the stylesheet bundle from 87 KB to 31 KB · Replaced 963 metadata blocks with one graph · Closed the colour system at 28 colours · Generated 495 pages from a SQLite export · Replaced 27 font sizes with a 6-step scale · Combined web development with SEO and content · Brand communication informed by audience response · A consistent identity across channels

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://engineer.company/portfolio/