**Engineer**

# Documentation

38 achievements

## Connect the parts of your system

The work collected under “Documentation” shows one part of a working system. We can help connect that part to the rest of your operation. Handovers, runbooks and manuals that let your own staff operate the system without ringing whoever built it.

**Explore our services** Get in touch →

## Find a comparable challenge

Follow an example that resembles your situation. Its review describes the decisions behind the result, beyond the technology label. One infrastructure plan for 14 departments — 14 research departments on one infrastructure plan

**One infrastructure plan for 14 departments** Get in touch →

### 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.

### 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.

### 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.

### 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.

### 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.

### Gathered and analyzed business requirements

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

### Improved team communication and collaboration

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

### 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.

### Authored 578 go-task automation targets

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

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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 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.

### 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.

### 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.

### 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.

### 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.

### 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.

## Focus the engineering effort

Bring the decision, the people affected and the options already considered. We can make the tradeoffs explicit and leave a documented recommendation.

**See what it costs** Get in touch →

# Put the subject in your context

Tell us how “Documentation” appears in your own system. We can help choose the relevant service and define the work around it.

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/categories/documentation/