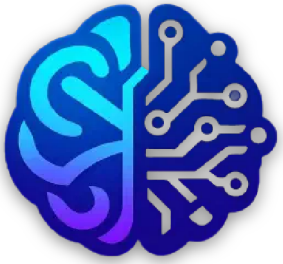
We run the platforms **your software ships on** — rebuilt from code, watched, and written down.

Cloud, infrastructure as code, CI/CD, containers, monitoring, backups, networks and access — from Copenhagen.



company
Engineer
Platform

Infrastructure you can rebuild from a repository

+45 50 10 61 84 · Copenhagen, Denmark

Carefully Crafted With Love by D.

Cloud infrastructure as code, automated deployment and recovery planning for the systems your business depends on. The work records describe the starting situation, what was done and the result.

Start with one deployment path or operational risk. We can agree a review or implementation with a documented rollout, rollback and ownership plan, so your team knows what to operate and how.

Infrastructure work · Pricing and engagement options · Discuss your project

Start with the result you need

Tell us what your business needs to achieve. We connect software, data and infrastructure around the people who use them.

Explore our services Get in touch →

See how we approach the work

Find a service that matches your goal and see examples of what was delivered and how.

Cases Get in touch →

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

See the whole catalogue

One partner for **10 services** from the first server to the 3 a.m. page.

Browse the catalogue

{ Cloud, Infrastructure & DevOps }

Where the software runs

Cloud Infrastructure & Migration

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.

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.

See all 9 achievements

How we work

Understand first

We learn the problem before we propose a solution. The brief you arrive with is the start of that conversation, not the end of it.

Build it to last

Durable over quick, clarity over cleverness. The system still has to be workable by the people who inherit it.

Stand behind it

We agree any ongoing operation or support as part of the engagement. Handover makes responsibilities and the next contact route clear.

{ Cloud, Infrastructure & DevOps }

Servers you can rebuild from versioned configuration

Infrastructure as Code

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

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.

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.

See all 17 achievements

{ Cloud, Infrastructure & DevOps }

Shipping changes without drama

DevOps & CI/CD Automation

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.

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.

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.

See all 43 achievements

{ Cloud, Infrastructure & DevOps }

Software that runs the same everywhere

Containerization & Orchestration

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

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.

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.

Built GitHub Actions CI/CD pipelines

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

See all 5 achievements

{ Cloud, Infrastructure & DevOps }

Knowing before your customers do

Site Reliability & Monitoring

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.

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.

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.

See all 23 achievements

{ Cloud, Infrastructure & DevOps }

Getting everything back after a bad day

Backup & Disaster Recovery

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

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.

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.

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.

See all 5 achievements

{ Cloud, Infrastructure & DevOps }

Keeping the lights on

System Administration

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

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.

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.

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.

See all 28 achievements

{ Cloud, Infrastructure & DevOps }

Connecting your offices and your people

Networking & VPN Setup

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

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.

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.

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.

See all 11 achievements

{ Cloud, Infrastructure & DevOps }

Deciding who can reach what

Security & Access Management

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

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.

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.

See all 31 achievements

{ Cloud, Infrastructure & DevOps }

Making sure your AI only touches what it should

AI Agent Security

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.

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.

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.

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.

See all 9 achievements

We work with **the platforms** your business already runs on.

Apple Microsoft DigitalOcean AWS Google

See the whole stack

Security review for the AI agents in scope

Mapped

An agreed inventory of the agents, credentials and data access in scope, kept with your configuration.

Locked

Each agent on its own identity and least privilege, and the text it reads treated as data, never as orders.

Recorded

Every action logged with the person it was for, where the agent cannot change it, and alerts on the unusual.

See how agents are locked Ask about your agents →

Tell us what you are building.

No form, no funnel, no sales sequence — just one e-mail to a person who will read it.

Start the conversation

This address shows the infrastructure work only. The company also builds software, works with data and advises teams — that side of it is at engineer.company.
See everything Engineer ApS does →

Begin with a focused improvement

A workflow, a data flow or a system that needs improvement can be a useful first task. We can agree its scope together.

See what it costs Get in touch →

Plan the first conversation

Tell us the result you need, what is already in place and when you need it. We can use that to discuss the next step.

Explore our services

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

On the server, we work with **open-source Linux** — no licences to pay for, and nothing hidden.

Ubuntu Debian

How we run servers

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://platform.engineer.company/