Engineer

Portfölj

Ingenjörstjänster med exempel på utfört arbete.

Engineer ApS registrerades 2025. Merparten av arbetet som visas här levererades dessförinnan av vår grundare inne i kundernas egna team; en del har levererats av företaget självt, och en del sedan dess i anställning. Samma händer och samma krav — vi säger vad som är vad i stället för att sudda ut det.

Läs om arbetet **i sitt sammanhang** — besluten, leveransen och de beskrivna resultaten.

Se hela katalogen

Hitta erfarenhet värd att använda

Använd översikten för att hitta erfarenhet som är relevant för ert projekt.

Se våra tjänster Hör av er →

Läs besluten bakom arbetet

Exemplen beskriver levererat arbete över flera tjänster. Läs en genomgång för att se situationen och besluten.

Fallstudier Hör av er →

Dessa namn **återkommer i arbetet nedan** — inget exotiskt som ert team måste ta över.

Python PostgreSQL TypeScript React GitHub Docker

Se alla tekniker

Bläddra efter kategori

MJUKVARUUTVECKLING

Backendutveckling Frontendutveckling Fullstackutveckling API:er och integration Webbutveckling Designsystem och UI UX/UI-design Test och kvalitetssäkring Prestandaoptimering Python Internationalisering

DATA & DATABASER

Databaser · PostgreSQL · SQL · Data engineering · Datapipelines (ETL/ELT) · Dataanalys · Datastyrning · Datalager · GIS / geodata

MOLN, INFRASTRUKTUR & DEVOPS

Molntjänster · Containrar (Docker/Kubernetes) · DevOps · Automatisering och CI/CD · Infrastruktur · Linux och servrar · Systemadministration · Nätverk och VPN · Övervakning och observerbarhet · Tillförlitlighet och säkerhetskopiering · Säkerhet · Plattformsarkitektur · Migrering och modernisering

RÅDGIVNING & LEDARSKAP

Tekniskt ledarskap · Teamledning · Mentorskap och coachning · Agilt arbete och Scrum · Projektledning · Produkt och krav · Intressenter och rapportering · Dokumentation · Lösningsarkitektur · Varumärke och marknadsföring

Allt nedan **är levererat** — inte föreslaget, inte planerat.

Varje post berättar vad arbetet förändrade och länkar till tjänsterna den visar.

Det vi erbjuder Vad det kostar →

{ Mjukvaruutveckling }

Plattforms- och lösningsarkitektur

Bestämmer hur helheten ska byggas

Systemets form utarbetad på papper först, så att delarna passar ihop och kostnaden är känd innan någon skriver kod.

Införde strikt samtidighet i Swift 6 ovanpå C

Införde fullständig strikt samtidighet i Swift 6 utan actors och överbryggade en blockerande händelseloop i C till main actor genom en producent, en konsument och en ordning — efter att ha konstaterat att en task per händelse förlorar den ordning som gränssnittet är beroende av.

Ett begränsat händelseflöde i en macOS-app

Begränsade händelsebuffertar och loggning efter en dokumenterad macOS-incident med en tillväxt på 41 MB i sekunden och 111 GB komprimerade sidor på en maskin med 36 GB; en dokumenterad fullständig testkörning minskade kärnans loggrader från 610 996 till 1 411.

Byggde en native macOS-klient i Swift 6

Byggde en native meddelandeklient för macOS i Swift 6 och SwiftUI — 11 141 rader i 53 filer — ovanpå C-gränssnittet till en Rust-kärna som länkas som ett statiskt arkiv från en låst revision.

Byggde in en QR-kodare på 522 rader

Byggde in en QR-kodare — Reed-Solomon över GF(256), fast modullayout, åtta maskmönster — på 522 rader i stället för att ta in ett tredje körtidsberoende, och verifierade den genom att läsa tillbaka den färdiga matrisen med en oberoende skriven avkodare.

Byggde en databasdriven CV-generator

Byggde en databasdriven generator för CV, referenser, portfolio och personliga brev i Python — 41 moduler, 10 580 rader — som renderar sex utdataformat från en SQLite-källa med 23 tabeller, sammanställd av en idempotent pipeline i 19 steg.

Offentliga tjänster på en liten server

Drev företagets offentliga tjänster på en värd med en kärna och 512 MB — en git-forge, en webbserver för sju domäner, Tor, två servrar för alternativa protokoll och intrångsblockering — med 464 MB användbart minne som designbegränsning; backupautomatik är en separat implementation.

Byggde hela kärninfrastrukturen från noll

Utformade och byggde, som en av de första anställda, hela kärninfrastrukturen och de stödjande processerna från noll för en SaaS-startup inom grön energi och lade grunden för snabb tillväxt.

Byggde infrastruktur som ännu körs år senare

Planerade och implementerade ny infrastrukturfunktionalitet för interna och externa system och byggde lösningar som höll tillräckligt väl för att fortfarande vara i drift flera år senare med minimala ändringar.

Fattade plattformens tekniska grundbeslut

Fattade plattformens grundläggande beslut under veckorna efter att repot öppnades i november 2025 — skiktningen, dataåtkomst med databasen först och ribban med noll varningar — och de håller fortfarande nio månader senare.

Modellerade 348 tabeller i 34 Postgres-scheman

Modellerade sjöfartsdomänen i 348 normaliserade tabeller i 34 PostgreSQL-scheman — yrkesverksamma, företag, fartyg, jobb, recensioner och resten — med SMALLINT-uppslag och UUID v7-nycklar.

Utformade ett funktionsbaserat skikt i PostgreSQL

Utformade ett funktionsbaserat dataskikt i PostgreSQL — 1 275 lagrade funktioner i 34 scheman — så att varje läsning och skrivning går via en funktion som databasen kan ge behörighet till, i stället för via en tabell.

Migrerade API:t till Huma v2 med 100 % paritet

Migrerade HTTP-API:t från Fiber till Huma v2 — 649 sökvägar och 760 operationer — och nådde och behöll 100 % paritet mellan de rutter som servern registrerar och den OpenAPI-beskrivning den publicerar.

Byggde växling av organisationskontext

Utformade ett system för att växla organisationskontext med localStorage hos klienten och en speglad cookie på serversidan, så att användare kan agera som de organisationer de förvaltar medan auktorisering enligt minsta behörighet upprätthålls.

Utformade en JSON-väg rakt från Postgres till API

Utformade en arkitektur för direkt vidarebefordran av JSON, där PostgreSQL-funktioner returnerar komplett JSON som Go-API:t skickar vidare ordagrant, vilket tog bort mellanliggande deserialisering och frikopplade frontenden från schemaändringar.

En maritim plattform med tydliga systemgränser

Utformade arkitekturen för en skiktad sjöfartsplattform som håller isär en PWA-frontend i Next.js, ett API i Go (Huma/Fiber) och ett funktionsskikt i PostgreSQL, med all affärslogik i databasen.

Förenklad infrastruktur utanför AWS

Omarbetade en internationell kunds infrastruktur genom att ta bort oanvända tjänster, anpassa kapaciteten och flytta arbetslaster från AWS. Grundaren beskriver en minskning av molnkostnaderna med en faktor 10 utan förlust av produktivitet.

En kartprodukt, buren hela vägen

Utformade, utvecklade och stödde infrastruktur, databearbetning och en kartapplikation under 2 år, med både programvaruleveranser och driftunderhåll.

Förnyade interna processer, sparade 8 000 timmar

Gjorde en genomgripande översyn av de interna processerna och sparade 8 000 timmar genom att förbättra mjukvaruarkitekturen, systemen och effektiviteten i schemaläggningen.

Ledde GIS-kartappen till 10 gånger högre intäkter

Ledde mjukvaruutvecklingen av en GIS-kartapplikation, ökade intäkterna 10 gånger och positionerade produkten som en primär datatillgång.

En infrastrukturplan för 14 avdelningar

Utformade ett heltäckande infrastrukturramverk för ett forskningsinstitut som berörde 14 avdelningar, med flexibla moduler, enhetliga datapipelines och strukturerade supportstrategier för ett långsiktigt införande.

{ Mjukvaruutveckling }

Backend- och API-utveckling

Motorn bakom din app eller webbplats

Den del användarna aldrig ser: där data lagras, vilka reglerna är och hur andra system får kommunicera med den.

Tog i bruk den oanvända halvan av kärnan

Tog i bruk den hälft av meddelandekärnan som applikationen aldrig hade använt — överföring av säkerhetskopior, försvinnande meddelanden, redigering och omsändning av meddelanden, verifierade inbjudningar, proxyer och krypteringspolicy — och körde varje test mot det riktiga biblioteket utan mockar.

Skrev en parser som verifierar varje FFI-anrop

Skrev en parser som läser den verkliga C-headern på 7 308 rader och verifierar varje anropsställe, varje enum-konstant och att varje klass som äger en pekare är final, efter att en handskriven platshållarheader hade låtit anrop till tre borttagna funktioner kompilera, länka och krascha.

Flyttade varje sträng till ett innehållsträd

Flyttade varje sträng som användarna ser ut ur Python till ett innehållsträd med 874 filer på tre språk, efter att ha hittat döda översättningar som ingen kunde se var döda och en kontroll som i tysthet bedömde en tredjedel av resultaten.

Aktiverade varje lintregel på felnivå

Aktiverade varje regel i Python-lintern på felnivå och arbetade igenom 1 815 fynd ned till noll, där vart och ett av de få undantagen har ett skriftligt skäl och två av dem backas upp av en kontroll i stället för en kommentar.

Gav generatorn 981 tester och 92 % täckning

Krävde 981 testfall och minst 92 % grentäckning av generatorn, med varningar behandlade som fel, och säkerställde idempotens genom att köra hela byggpipelinen två gånger från en tom fil och kräva att den andra körningen inte ändrade något.

Byggde rekryterings- och karriärytor för sjöfolk

Byggde arbetsytor för rekrytering och sjöfolks karriär; den dokumenterade implementationsbilden omfattar 151 lagrade funktioner i 48 tabeller för tjänster, ansökningar, certifikat, rangutveckling och sjötid.

Byggde socialt skikt: inlägg, flöde och grupper

Byggde plattformens sociala skikt — inlägg, flöde, grupper, omnämnanden, en följargraf och en sammanställning av bemärkelsedagar — på samma funktionsbaserade dataskikt som resten av produkten.

Byggde en lojalitetsplånbok, ligor och märken

Byggde lojalitets- och ryktessystemet — 67 funktioner över en huvudbok i 31 tabeller, med ligor, märken och en inlösenbutik — som tar ett radlås på saldot för att stänga fönstret för dubbelspendering.

Byggde hastighetsbegränsning som stänger vid fel

Byggde skydd mot missbruk som stänger vid fel — 22 hastighetsbegränsare i Redis, Cloudflare Turnstile, idempotenta anrop och ett ursprungslås — så att plattformen avvisar botar och anropsfloder i stället för att lita på sina anropare.

Byggde egen felövervakning och spårning

Byggde egen felövervakning och OpenTelemetry-spårning i stället för att köpa dem — sanering av nyttolaster, upptäckt av toppar och regressioner, symbolisering och ett syntetiskt hjärtslag — bakom 11 operatörsvyer.

Flyttade långsamt arbete till en jobbkö i River

Flyttade långsamt arbete från anropsvägen till en jobbkö i River — 15 workermoduler, 8 schemalagda uppgifter och 20 pg_cron-jobb — så att ett anrop returnerar medan arbetet bakom det fortsätter.

Byggde åtkomsträttigheter via Stripe och appköp

Byggde skiktet för betalningar och åtkomsträttigheter — Stripe vid sidan av köp i appen via Apple och Google — som spärrar katalogen, sökningen och exporten bakom en åtkomstmodell i 11 tabeller som kontrolleras på servern.

Byggde en testsvit för Go, SQL och webbläsare

Byggde en skiktad automatiserad testsvit — 981 Go-tester, 543 specifikationer för frontend och webbläsare, 494 beteendetester i SQL — med mutationstestning, egenskapsbaserade tester och en tillgänglighetsgrind.

Byggde och underhöll Django-appar för IPTV

Utvecklade och underhöll Django-webbapplikationer för IPTV-plattformen, levererade nya funktioner och förbättrade prestanda och stabilitet.

Härdade applikationen med nonce-baserad CSP

Härdade applikationen med nonce-baserad CSP, HSTS, SameSite-cookies, databasroller med minsta behörighet och förnyade kontroller av åtkomsträttigheter på serversidan.

Implementerade en katalogstyrd djupsammanslagning

Implementerade en katalogstyrd djupsammanslagning av lagrade JSON-inställningar som förhindrar krascher på grund av saknade nycklar när schemat utvecklas.

Byggde e-post med failover och leveransloggning

Byggde e-post som en förmåga i plattformen — tre leverantörer med failover, webhooks för leveranser, loggning av utskick och leveranser, mallar och kampanjer — bakom en startkontroll som vägrar starta utan en leverantör.

Genererade API-kontraktet utåt från databasen

Genererade API-kontraktet utåt från databasen — OpenAPI, en typad TypeScript-klient på 44 076 rader, 61 mock-hanterare och de gränser som gränssnittet upprätthåller — med en spärr i varje led som fäller vid avvikelse.

Utvecklade ett rapportsystem för dataanalys

Utvecklade ett rapportsystem för dataanalys som ökade de kvartalsvisa mjukvaruintäkterna med 400 % genom Python-baserade PDF-rapporter.

Ledde över 30 GIS-projekt från bygge till support

Ledde utveckling, driftsättning och support av över 30 GIS-projekt och visade expertis inom PostgreSQL, Bash, Python, JavaScript, GDAL, ArcGIS, PostGIS och Mapbox.

Styrde full stack-utvecklingen av GIS-kartor

Styrde full stack-utvecklingen av GIS-kartor med ansvar för PostgreSQL, Mapbox, ReactJS och NodeJS för att leverera en integrerad lösning.

Levererade 8 Power BI-projekt med handböcker

Levererade 8 Power BI-projekt med utförliga handböcker och integrerade Microsoft Power BI-verktygen med ett NodeJS-API och Python FastAPI för effektiv dataanalys och visualisering.

Utformade och drev 6 ETL/ELT-pipelines

Utformade, implementerade och administrerade 6 ETL/ELT-pipelines med Google BigQuery, MSSQL, PostgreSQL, Shell-skript, PL/pgSQL och Transact-SQL och integrerade data för effektiv bearbetning i ett Python-API.

Tjänsten ingår i arbete som står annanstans på sidan:

Införde strikt samtidighet i Swift 6 ovanpå C · Byggde in en QR-kodare på 522 rader ·

Byggde en databasdriven CV-generator · Fattade plattformens tekniska grundbeslut ·

Utformade ett funktionsbaserat skikt i PostgreSQL · Migrerade API:t till Huma v2 med 100 % paritet ·

Byggde växling av organisationskontext · Utformade en JSON-väg rakt från Postgres till API ·

En maritim plattform med tydliga systemgränser

{ Mjukvaruutveckling }

Frontendutveckling

Skärmarna som människor faktiskt använder

Gränssnitt som laddar snabbt, fungerar i mobilen och beter sig likadant i alla webbläsare dina kunder råkar använda.

Byggde ett designsystem för macOS med 48 tokens

Byggde ett designsystem med 48 tokens på plattformens eget glasmaterial och skrev lintern som avvisar ett magiskt tal, en hårdkodad teckenstorlek eller en animering som ignorerar inställningen för reducerad rörelse.

Gav alla 17 ikoner namn för skärmläsare

Gav varje kontroll i gränssnittet som bara består av en ikon ett namn för skärmläsare efter att ha hittat skicka-knappen uppläst som ”arrow up circle, button”, och skrev lintern som kräver etiketten inom åtta rader från ikonen.

Lintade själva kvalitetsgrinden på 10 242 rader

Förde in 10 242 rader JavaScript för kvalitetsgrinden under en formaterare och en linter efter att ha konstaterat att det var den största kodmassan i repot och den enda som inget läste, och åtgärdade 13 fynd utan att undertrycka något.

Krympte stilmallspaketet från 87 KB till 31 KB

Krympte stilmallspaketet från 87 KB till 31 KB genom att ta ut ett base64-kodat typsnitt ur det, och tog bort 756 KB i 22 filer som publicerades vid varje driftsättning utan att något refererade till dem.

Byggde en utskriftsstilmall på 637 rader

Byggde en utskriftsstilmall på 637 rader mot fyra dokumenterade beteenden i renderingsmotorer, bland annat run-in-rubriker som skrevs ut vitt på vitt för varje läsare vars webbläsare föredrog mörkt läge.

Fick reduced-motion att hålla vad den lovar

Fick reduced-motion att hålla vad den lovar efter att ha hittat elva selektorer som fortfarande animerade under inställningen, eftersom en universell transition-none-regel förlorar på specificitet mot varje regel som har en klass.

Fann att det ljusa temat saknade ett helt skikt

Upptäckte att det ljusa temat hade saknat ett helsidesöverlägg sedan den dag det skrevs, genom att kräva att båda teman målar varje skiktad yta med samma antal skikt.

Stängde färgsystemet vid 28 färger

Stängde färgsystemet vid 28 dokumenterade färger med en linter som fäller ett värde som målas men inte är dokumenterat, dokumenteras men inte målas, är felmätt, skrivs om som en literal eller ligger inom ett perceptuellt avstånd på 0,02 från en befintlig färg.

Tillgänglighetsfelen som ert verktyg missar

Hittade och åtgärdade 11 tillgänglighetsbrister som webbläsaren rapporterade som felfria — åtta sidfotslänkar som låg kvar i tabbordningen bakom pointer-events, en scroll-tidslinje som klippte bort kolofonen på tre sidor och en regelmotor som körde 70 av sina 105 regler.

Tillgänglighetskontroller på 32 sidor i båda teman

Implementerade WCAG 2.2 Level AA-kontroller på 32 representativa sidor i båda färgteman, med en överensstämmelseförteckning, kontroll av ett AAA-kriterium och det partiella resultatet för kriterium 1.4.13 uttryckligen angivet.

Ersatte 27 teckenstorlekar med en skala i 6 steg

Ersatte 27 renderade teckenstorlekar, vars närmaste grannar låg 0,6 % från varandra, med en Major Third-skala i sex steg, och skrev lintern som fäller den tjugoåttonde.

Ett skyddsnät som inte längre kostar en halvtimme

Kortade webbplatsens webbläsarstyrda kvalitetsgrind från 1 636 sekunder till 615 genom att schemalägga kontrollerna med den längsta först i en workerpool begränsad till fyra spår, efter att ha mätt att alfabetisk ordning kostade 320 sekunder mot 224.

Byggde en flerspråkig Hugo-webbplats på 3 språk

Byggde en flerspråkig statisk webbplats i Hugo med 108 mallfiler, varav 53 partials, som publicerar varje sida i fyra representationer från ett innehållsträd på tre språk, och en gång till under sju fokuserade underdomäner byggda från samma träd.

Byggde plattformens backoffice för operatörer

Byggde operatörernas backoffice — ca 40 administrationsrutter och 128 komponenter för anspråk, moderering, funktionsflaggor, cache och diagnostik — så att plattformen kan drivas utan databasåtkomst.

Skötte Hugo-webbplats, förbättrade produktens UI

Underhöll och vidareutvecklade den äldre statiska webbplatsen i Hugo och bidrog samtidigt med UI/UX-förbättringar till den huvudsakliga produkten för tillgångsförvaltning.

Byggde ett i18n-system för engelska och danska

Etablerade ett internationaliseringssystem för engelska och danska med 12 027 meddelanden per språkversion i 458 namnrymdsfiler, med ett ordförråd som lintern upprätthåller och en budget på 400 rader per fil.

Utformade hela produktens UI och UX

Utformade produktens UI och UX från början till slut — katalogrutnät, dubbla kort- och tabellvyer, kravvalidering i realtid och brödsmulenavigering.

Byggde ett ”Liquid Glass”-designsystem för iOS 26

Utformade ett designsystem i stil med ”Liquid Glass” från iOS 26 och en kanonisk komponentförteckning som lintregler upprätthåller för att förhindra att gränssnittet spretar.

Levererade fullt stöd för Progressive Web App

Levererade fullt stöd för Progressive Web App — installerbar och med offlinestöd — med körtidscachning i Workbox via next-pwa.

Byggde Next.js 16-frontenden: SSR, SSG och CSR

Byggde frontenden i Next.js 16 med medvetna SSR-, SSG- och CSR-strategier och en återanvändbar fabrik för förhämtade serversidor som tar bort N+1-kaskaden av hämtningar från varje inloggad sida som flyttas över till den.

Utformade 30 webbplatser från Photoshop till HTML

Utformade 30 webbplatser och levererade unika och flexibla lösningar genom att omvandla Photoshop-designer till HTML.

Utvecklade 100 webbapplikationer med HTML/HTML5

Utvecklade 100 webbapplikationer med HTML/HTML5, CSS/SCSS, Django, WordPress och Joomla och säkerställde en mångsidig närvaro på nätet.

Bidrog till designen av användargränssnitt

Bidrog till designprocesserna för användargränssnitt och säkerställde intuitiva, visuellt tilltalande och användarvänliga gränssnitt i projekten.

Presenterade 200 UI/UX-förbättringar

Presenterade 200 UI/UX-förbättringar för GIS-kartapplikationen och ökade intäkterna 10 gånger genom förbättrade programfunktioner.

Tjänsten ingår i arbete som står annanstans på sidan:

Ett begränsat händelseflöde i en macOS-app · Byggde en native macOS-klient i Swift 6 ·

Fattade plattformens tekniska grundbeslut · Byggde växling av organisationskontext ·

En maritim plattform med tydliga systemgränser · Styrde full stack-utvecklingen av GIS-kartor

{ Mjukvaruutveckling }

Fullstackutveckling av produkter

En fungerande produkt, hela vägen

Ett team som tar en idé från en skiss till något dina kunder kan logga in i, i stället för att lämna den vidare mellan tre leverantörer.

Byggde anspråk på företagsägarskap och prövning

Byggde anspråk på företagsägarskap från början till slut — en användare gör anspråk på ett företag, en administratör avgör, och ett godkännande skriver om auktoriseringsgrafen som bestämmer vem som får redigera vad.

Tjänsten ingår i arbete som står annanstans på sidan:

Byggde en native macOS-klient i Swift 6 · Byggde en databasdriven CV-generator ·

Byggde en flerspråkig Hugo-webbplats på 3 språk · Byggde rekryterings- och karriärytor för sjöfolk ·

Byggde socialt skikt: inlägg, flöde och grupper · Byggde plattformens backoffice för operatörer ·

Byggde och underhöll Django-appar för IPTV · Fattade plattformens tekniska grundbeslut · Byggde Next.js 16-frontenden: SSR, SSG och CSR · En maritim plattform med tydliga systemgränser · En kartprodukt, buren hela vägen · Utvecklade 100 webbapplikationer med HTML/HTML5 · Ledde över 30 GIS-projekt från bygge till support · Styrde full stack-utvecklingen av GIS-kartor · Ledde GIS-kartappen till 10 gånger högre intäkter

{ Mjukvaruutveckling }

Webbutveckling och CMS

En webbplats du kan uppdatera själv

En webbplats som är snabb, lätt att hitta och går att redigera av din egen personal, utan en faktura för varje ändrad formulering.

Publicerade en läsbar onion-spegel i Tor

Publicerade webbplatsen som en onion-spegel i Tor på en läsbar adress som börjar med engineer, med nyckeln genererad utanför värden så att den aldrig nådde detta repo eller en bärbar dator, och lät den förbli den enda av fyra ytterdörrar som aldrig räknas.

Byggde den visuella identiteten ur 2 teckningar

Byggde företagets ikon- och favicon-uppsättningar från två handgjorda teckningar, där varje härledd resurs genereras om med skript och en kontroll fäller när en härledd fil har committats före sin källa.

Klarade 106 av 106 PDF/UA-1-profilkontroller

Dokumenterade PDF/UA-1-validering av nio dokument med 106 av 106 godkända kontroller i den konfigurerade tillgänglighetsprofilen, medan arkivvarianten underkändes på en av 146 regler.

Rättade en sitemap med en enda delad tidsstämpel

Rättade en sitemap där 172 av 176 URL:er delade samma ändringstidsstämpel genom att hämta datumet ur git-historiken, efter att ha konstaterat att exporten skriver om varje fil vid varje körning.

Ersatte 963 metadatablock med en enda graf

Ersatte 963 anonyma block av strukturerad data som upprepade företaget 1 671 gånger med en enda länkad graf av 16 typer, präglade från stabila ursprungsidentifierare.

Speglade webbplatsen till Gemini och Gopher

Byggde Gemini- och Gopher-utdata från samma webbkälla; resultatbeskrivningen anger 777 dokument i varje format vid det skedet och noll ändrade HTML-byte genom tillägget av formaten.

Genererade 495 sidor från en SQLite-export

Genererade 495 resultatsidor på tre språk från en skrivskyddad SQLite-export, med sidadressen angiven som data så att en rättad mening inte längre flyttade sidan och bröt länken.

Er kod på er egen server

Driftsatte företagets egen git-forge på Soft Serve, privat som standard, utan webbpanel och med SSH-porten bunden till loopback bakom en hoppserver, och gjorde landningssidan framför den till en byggartefakt från huvudwebbplatsen i stället för en manuellt underhållen kopia.

Kombinerade webbutveckling med SEO och innehåll

Drev kundernas framgång på webben genom att kombinera skräddarsydd webbutveckling och design med SEO, innehållsstrategi och copywriting.

Administrerade 40 webbplatser på Apache och Nginx

Administrerade 40 webbplatser på värdservrar med Ubuntu Linux, Apache och Nginx och säkerställde hög tillgänglighet och prestanda.

Tjänsten ingår i arbete som står annanstans på sidan:

Krympte stilmallspaketet från 87 KB till 31 KB · Byggde en utskriftsstilmall på 637 rader · Fick reduced-motion att hålla vad den lovar · Tillgänglighetsfelen som ert verktyg missar · Tillgänglighetskontroller på 32 sidor i båda teman · Byggde en flerspråkig Hugo-webbplats på 3 språk · Skötte Hugo-webbplats, förbättrade produktens UI · Byggde och underhöll Django-appar för IPTV · Levererade fullt stöd för Progressive Web App · Byggde Next.js 16-frontenden: SSR, SSG och CSR · Utformade 30 webbplatser från Photoshop till HTML · Utvecklade 100 webbapplikationer med HTML/HTML5

{ Mjukvaruutveckling }

Redo för AI-agenter

Syns när kunderna frågar en AI

En webbplats eller app som de AI-assistenter dina kunder använder kan hitta, läsa och agera på, kontrollerad vid varje release i stället för utlovad en gång.

Drev utvecklingen som 51 skriftliga initiativ

Drev webbplatsens utveckling som 51 skriftliga initiativ i 54 plandokument, vart och ett med sina öppna rutor, sina mätningar och de beslut det avböjde — bland dem sex av åtta fynd om agentberedskap som avvisades med skälet nedskrivet.

Tjänsten ingår i arbete som står annanstans på sidan:

Rättade en sitemap med en enda delad tidsstämpel · Ersatte 963 metadatablock med en enda graf · Tillgänglighetskontroller på 32 sidor i båda teman · Byggde en flerspråkig Hugo-webbplats på 3 språk

{ Mjukvaruutveckling }

Verktyg för AI-agenter och MCP-servrar

Låta AI använda programvaran du redan har

Den programvara du redan har, öppnad som verktyg en AI-agent kan anropa, utan att något som redan fungerar behöver skrivas om.

Tjänsten ingår i arbete som står annanstans på sidan:

Byggde plattformens backoffice för operatörer · Flyttade långsamt arbete till en jobbkö i River · Genererade API-kontraktet utåt från databasen · Migrerade API:t till Huma v2 med 100 % paritet · Utformade en JSON-väg rakt från Postgres till API

{ Mjukvaruutveckling }

GIS- och geodatalösningar

Kartor som besvarar frågor

Allt som händer på en plats — fartyg, fordon, tomter, sensorer — på en karta du kan söka i, mäta på och dela.

Ledde 30 framgångsrika GIS-projekt till leverans

Ledde genomförandet av 30 framgångsrika GIS-projekt och visade ledarskap genom att leverera innovativa lösningar inom hela området.

Optimerade prognos- och investeringsstrategier

Optimerade prognos- och investeringsstrategier för 11 elnätsföretag och drev på driftseffektiviteten med datadrivna GIS-lösningar.

Automatiserade leveransen av 20 GIS-datapipelines

Automatiserade leveransen av 20 GIS-datapipelines och ETL-processer för applikationsdata, vilket effektiviserade infrastrukturautomatiseringen och rapporteringen.

Automatiserade GIS-driftsättning i GitHub Actions

Automatiserade driftsättningen av en GIS-applikation som SaaS, databearbetningen och rapporteringssystemet med GitHub Actions CI/CD, Python, Bash och SQL.

Drev 100 datalager i PostgreSQL, MS SQL, BigQuery

Utformade, byggde upp och förvaltade 100 datalagerdatabaser i PostgreSQL, MS SQL och Google BigQuery, främst med GIS- och tidsseriedata, och optimerade prestanda och skalbarhet.

Publicerade 500 rapporter om el- och GIS-data

Publicerade 500 analysrapporter om el- och GIS-data och använde grundlig research och felsökning för att säkerställa korrekta insikter ur stora mängder geodata och tidsseriedata.

Löste 1 000 problem i geodata och tidsseriedata

Löste 1 000 problem i geodata och tidsseriedata med GDAL, ArcGIS, PostGIS, Mapbox, QGIS, SQL (PL/pgSQL, Transact-SQL) och Bash, vilket säkrade en högkvalitativ bearbetning av stora datamängder.

Gjorde GIS-kartan 10 gånger snabbare i PostgreSQL

Förbättrade prestandan i en kartapplikation för geodata 10 gånger genom en strategisk databasövergång från MSSQL till PostgreSQL, inklusive själva omställningen av data, och optimerade bearbetningen och datasäkerheten.

Gjorde pipelinen för geodata 50 gånger snabbare

Dokumenterade en förbättring på 50 gånger i en geografisk datapipeline i grundarens arbete hos Utiligize genom SQL- och datamodelländringar i PostgreSQL, MS SQL och Google Cloud BigQuery.

Tjänsten ingår i arbete som står annanstans på sidan:

En kartprodukt, buren hela vägen · Ledde över 30 GIS-projekt från bygge till support · Styrde full stack-utvecklingen av GIS-kartor · Ledde GIS-kartappen till 10 gånger högre intäkter

{ Mjukvaruutveckling }

UI/UX-design och designsystem

Gör det självklart hur det används

Layout, formuleringar och flöde ordnade så att människor gör klart det de kom för i stället för att ringa din kundtjänst.

Skrev om felmeddelanden som skyllde på användare

Skrev om applikationens felmeddelanden enligt en skriftlig tonstandard efter att en nekad inloggning hade skyllt på användaren för felskrivning när leverantören i själva verket krävde ett appspecifikt lösenord, och täckte det med ett test som namnger leverantören.

Tjänsten ingår i arbete som står annanstans på sidan:

Byggde den visuella identiteten ur 2 teckningar · Byggde ett designsystem för macOS med 48 tokens · Gav alla 17 ikoner namn för skärmläsare · Byggde en native macOS-klient i Swift 6 · Klarade 106 av 106 PDF/UA-1-profilkontroller · Byggde en utskriftsstilmall på 637 rader · Fick reduced-motion att hålla vad den lovar · Fann att det ljusa temat saknade ett helt skikt ·

Stängde färgsystemet vid 28 färger · Tillgänglighetsfelen som ert verktyg missar · Tillgänglighetskontroller på 32 sidor i båda teman · Ersatte 27 teckenstorlekar med en skala i 6 steg · Kombinerade webbutveckling med SEO och innehåll · Skötte Hugo-webbplats, förbättrade produktens UI · Utformade hela produktens UI och UX · Byggde ett ”Liquid Glass”-designsystem för iOS 26 · Utformade 30 webbplatser från Photoshop till HTML · Bidrog till designen av användargränssnitt · Presenterade 200 UI/UX-förbättringar

{ Mjukvaruutveckling }

Internationalisering och lokalisering

Fungerar på mer än ett språk

Mjukvara som känns naturlig på varje marknad du säljer på, inklusive datum, valutor och hur namn sorteras.

Byggde en sifferkontroll mellan språken

Byggde en innehållskontroll mellan språken som fäller när en översättning tappar en siffra som den engelska texten anger, och när ett språk använder en notation som det inte använder — och hittade två danska beskrivningar som saknade ett mätvärde och sexton ukrainska textavsnitt som citerade på engelskt sätt.

Tjänsten ingår i arbete som står annanstans på sidan:

Flyttade varje sträng till ett innehållsträd · Byggde en databasdriven CV-generator · Speglade webbplatsen till Gemini och Gopher · Genererade 495 sidor från en SQLite-export · Tillgänglighetskontroller på 32 sidor i båda teman · Byggde en flerspråkig Hugo-webbplats på 3 språk · Byggde ett i18n-system för engelska och danska

{ Mjukvaruutveckling }

Utveckling av skräddarsydda AI-agenter

En AI-assistent som kan din bransch

En agent kopplad till de system du redan använder,
testad på fall som dina egna experter skriver, övervakad
av dina medarbetare och aldrig låst till en enda modell.

Tjänsten ingår i arbete som står annanstans på sidan:

Fick sex insamlare att ange vad de hoppade över · Skrev tester för själva kontrollerna ·
Byggde en commit-grind med 22 linters · Byggde egen felövervakning och spårning ·
Byggde repots kontrollmotor med 268 kontroller

{ Data & databaser }

Databasdesign och datamodellering

Ett vettigt ställe att förvara dina data

En struktur som speglar hur verksamheten faktiskt fungerar, så
att de frågor du behöver svar på senare fortfarande går att ställa.

Byggde egen analys inuti PostgreSQL

Byggde egen produktanalys i PostgreSQL — 47 funktioner över partitionerade händelsetabeller som rensar sig själva — pseudonymiserad bakom ett roterande salt och villkorad av besökarens samtycke.

Tidsordnade nycklar för en växande datamodell

Införde tidsordnade UUID v7-identifierare (PostgreSQL 18) som entitetsnycklar för att minska fragmenteringen i B-trädindex och snabba upp databasfrågor.

Tjänsten ingår i arbete som står annanstans på sidan:

Byggde en databasdriven CV-generator · Byggde rekryterings- och karriärytor för sjöfolk ·
Byggde en lojalitetsplånbok, ligor och märken · Flyttade långsamt arbete till en jobbkö i River ·
Byggde åtkomsträttigheter via Stripe och appköp · Modellerade 348 tabeller i 34 Postgres-scheman ·
Implementerade en katalogstyrd djupsammanslagning ·

Utformade ett funktionsbaserat skikt i PostgreSQL ·

Utformade en JSON-väg rakt från Postgres till API · En maritim plattform med tydliga systemgränser ·

Drev 100 datalager i PostgreSQL, MS SQL, BigQuery · Utformade och drev 6 ETL/ELT-pipelines

{ Data & databaser }

Databasadministration (DBA)

Någon som tar hand om databasen

Någon som ansvarar för det som hela företaget vilar på: den är i drift, den är säkerhetskopierad och den förfaller inte i det tysta.

1 022 migreringar bakom en grind mot schemadrift

Höll schemat korrekt genom 1 022 migreringar med en CI-grind som bygger databasen på båda sätten — en ny installation, och en installation plus varje migrering — och fäller när de två skiljer sig åt.

Byggde säkerhetskopior och återställning som kod

Konfigurerade lagringstiden för databasens säkerhetskopior som infrastruktur som kod, granskade sedan återställningsförmågan och dokumenterade rutinen för återläsning — och pekade ut de återstående luckorna i stället för att låta dem upptäckas under en incident.

Byggde 600 ETL/ELT-pipelines i PL/pgSQL

Byggde 600 PL/pgSQL-baserade ETL/ELT-pipelines för att effektivisera komplexa arbetsflöden för databearbetning i flera utvecklings- och produktionsmiljöer för PostgreSQL.

Automatiserade databearbetning i Python, PL/pgSQL

Automatiserade uppgifter inom databearbetning med Shell-skript, PL/pgSQL, Python och Transact-SQL och ökade produktiviteten och effektiviteten.

Automatiserade 100 kritiska säkerhetskopior

Automatiserade 100 kritiska säkerhetskopior av data med Barman, Google Cloud, Bash och Python och säkerställde dataintegriteten i samtliga databaser.

Gjorde PostgreSQL 10 gånger snabbare

Gjorde PostgreSQL 10 gånger snabbare med strategisk indexering, partitionering och optimering av databasfrågor och effektiviserade databasen för användar- och tenantdata, geospatiala data och tidsseriedata om el.

Skrev 1 000 RBAC-regler som säkrar Linux-servrar

Stärkte datasäkerheten genom att införa 1 000 RBAC-regler för utvecklare, applikationsinstanser, PostgreSQL, MS SQL och andra Linux-servrar, vilket förhindrade obehörig åtkomst; dokumenterade reglerna som automatisering i Ansible.

Drev 10 PostgreSQL- och MS SQL-servrar på Ubuntu

Utformade, driftsatte och underhöll 10 PostgreSQL- och MS SQL-servrar på VPS:er med Ubuntu Linux och säkerställde optimal serverprestanda och tillförlitlighet.

Tjänsten ingår i arbete som står annanstans på sidan:

Drev 100 datalager i PostgreSQL, MS SQL, BigQuery

{ Data & databaser }

Optimering av databasprestanda

Gör de långsamma delarna snabba

Hittar databasfrågorna och de saknade indexen bakom ett system som har blivit tyngre för varje år, och åtgärdar dem som spelar roll.

30 dagars eldata på 180 ms

Utvecklade varianter för timvis aggregering av elförbrukning i Python, SQL och Bash + Jq; Python-varianten behandlade en JSONL-datamängd för 30 dagar på 180 ms i prestandatestet.

Tjänsten ingår i arbete som står annanstans på sidan:

Tidsordnade nycklar för en växande datamodell · Gjorde PostgreSQL 10 gånger snabbare ·

Drev 100 datalager i PostgreSQL, MS SQL, BigQuery ·

Gjorde GIS-kartan 10 gånger snabbare i PostgreSQL ·

Gjorde pipelinen för geodata 50 gånger snabbare

{ Data & databaser }

Databasmigrering och modernisering

Bort från det gamla systemet

Data som flyttas ur något åldrande och in i något som fortfarande underhålls, medan verksamheten fortsätter som vanligt under tiden.

Tjänsten ingår i arbete som står annanstans på sidan:

Flyttade varje sträng till ett innehållsträd · 1 022 migreringar bakom en grind mot schemadrift · Gjorde GIS-kartan 10 gånger snabbare i PostgreSQL

{ Data & databaser }

Utveckling av datapipelines (ETL/ELT)

Data som tar sig från A till B av sig själv

Siffror som kommer fram dit de behövs, på fasta tider, utan att någon kopierar ett kalkylblad i slutet av månaden.

Sparade 4 000 timmar med GitHub och GitLab CI/CD

Effektiviserade processerna för dataanalys och mjukvaruutveckling och sparade 4 000 timmar genom att införa CI/CD-arbetssätt med GitHub, GitLab, Bash och Python.

Tjänsten ingår i arbete som står annanstans på sidan:

Genererade 495 sidor från en SQLite-export · Byggde egen analys inuti PostgreSQL · En kartprodukt, buren hela vägen · Byggde 600 ETL/ELT-pipelines i PL/pgSQL · Automatiserade databearbetning i Python, PL/pgSQL · Ledde över 30 GIS-projekt från bygge till support · Automatiserade leveransen av 20 GIS-datapipelines · Automatiserade GIS-driftsättning i GitHub Actions · Utformade och drev 6 ETL/ELT-pipelines · Löste 1 000 problem i geodata och tidsseriedata · Gjorde pipelinen för geodata 50 gånger snabbare · 30 dagars eldata på 180 ms · En infrastrukturplan för 14 avdelningar

{ Data & databaser }

Design av datalager

Ett ställe där alla siffror stämmer överens

Siffror från försäljning, drift och ekonomi samlade på ett ställe, så att två avdelningar slutar komma med två olika svar till samma möte.

Tjänsten ingår i arbete som står annanstans på sidan:

Drev 100 datalager i PostgreSQL, MS SQL, BigQuery · Utformade och drev 6 ETL/ELT-pipelines · Gjorde pipelinen för geodata 50 gånger snabbare

{ Data & databaser }

Dataanalys och BI-dashboards

Se vad som händer

Dashboards och rapporter som visar de få siffror ett beslut faktiskt hänger på, uppdaterade utan att någon behöver be om det.

Utformade ett tids- och rapporteringssystem

Utformade ett system för tidshantering och rapportering som användes i produktion i flera år utan större ändringar.

Levererade analyser och lägesrapporter till vd:n

Levererade analyser och regelbundna lägesrapporter till vd:n och omsatte tekniska nyckeltal och leveransstatus i beslut.

Byggde företagets första observerbarhetsdashboard

Utvecklade och lanserade företagets första dashboard för observerbarhet, med realtidsinsikt i systemens prestanda och datavisualisering på den stora tv:n på kontoret.

Varumärkeskommunikation utifrån publikens respons

Använde marknadsundersökningar och publikens återkoppling i Engineer ApS varumärkeskommunikation och uppsökande arbete.

Tjänsten ingår i arbete som står annanstans på sidan:

Byggde egen analys inuti PostgreSQL · Utvecklade ett rapportsystem för dataanalys · Sparade 4 000 timmar med GitHub och GitLab CI/CD · Optimerade prognos- och investeringsstrategier · Publicerade 500 rapporter om el- och GIS-data · Levererade 8 Power BI-projekt med handböcker · 30 dagars eldata på 180 ms

{ Data & databaser }

Datastyrning och datakvalitet

Kunna lita på siffrorna

Regler och kontroller som fångar dålig data på vägen in, och ett tydligt svar på vem som äger vilken siffra och var den kommer ifrån.

Mätte 13 adresser helt utan spårning

Mätte tretton adresser enbart utifrån webbserverns egen åtkomstlogg — ingen cookie, ingen spårare och ingen tjänst från tredje part — maskerade klientens nätverk när varje rad skrivs och förde in resultatet i ett permanent arkiv med 21 dagliga räkningar och 20 månatliga fasetter.

Känn till licensen för allt ni levererar

Granskade licenskompatibiliteten hos 644 Rust-crates vid varje bygge och bevisade att kontrollen slår till genom att skriva om licensen för en crate och genom att flytta den låsta kärnrevisionen utan att generera om.

Skrev tester för själva kontrollerna

Skrev tester för själva kontrollerna efter att ha konstaterat att en kontroll som bara matas med felfri indata en dag rapporterar felfritt för att den inte läste något — lade in en avsiktlig felstavning för att bekräfta att stavningskontrollen hittar den och hämtade ett id-intervall från databasen i stället för från ett tal i testet.

Skrev en avgränsningsregel efter läckta uppgifter

Skrev in en avgränsningsregel i repot efter att en omstrukturering hade fört in ett annat företags värdförteckning, brandväggsundantag och text i det — och höll de karantänsatta resterna kvar under skanningen efter hemligheter i stället för att undanta dem.

Samlade in och analyserade verksamhetskrav

Samlade in och analyserade verksamhetskrav och omsatte dem i genomförbara funktioner och user stories i linje med standarderna för datastyrning.

Tjänsten ingår i arbete som står annanstans på sidan:

Byggde en sifferkontroll mellan språken · Flyttade varje sträng till ett innehållsträd ·

Ersatte 963 metadatablock med en enda graf · Byggde egen analys inuti PostgreSQL ·

Modellerade 348 tabeller i 34 Postgres-scheman ·

Implementerade en katalogstyrd djupsammanslagning ·

Utformade ett funktionsbaserat skikt i PostgreSQL · Byggde 600 ETL/ELT-pipelines i PL/pgSQL ·

Publicerade 500 rapporter om el- och GIS-data · Löste 1 000 problem i geodata och tidsseriedata ·

En infrastrukturplan för 14 avdelningar

{ Data & databaser }

Dataförberedelse för AI

Data som en AI faktiskt kan använda

Dina data samlade där AI-agenter kan nå dem, med en enda version av varje kund, utan att någon ser mer än de får, och med varje svar spårat tillbaka till källan.

Tjänsten ingår i arbete som står annanstans på sidan:

Ersatte 963 metadatablock med en enda graf · Genererade 495 sidor från en SQLite-export · 1 022 migreringar bakom en grind mot schemadrift · Modellerade 348 tabeller i 34 Postgres-scheman · Utformade och drev 6 ETL/ELT-pipelines · Gjorde pipelinen för geodata 50 gånger snabbare

{ Moln, infrastruktur & DevOps }

Molninfrastruktur och migrering

Där mjukvaran körs

Servrar dimensionerade efter vad du faktiskt använder, flyttade till molnet eller tillbaka ut ur det, med månadskostnaden uträknad först.

Provisionerade en andra server från kod

Automatiserade provisioneringen av en andra server hos en andra molnleverantör och skapade brandväggen före maskinen så att den föds bakom en, med båda leverantörernas brandväggar skrivna direkt mot deras REST-API:er för att undvika en collection från tredje part.

Körde containrar på Podman och Quadlet

Driftsatte en containermiljö på Podman och Quadlet under systemd i stället för Docker, eftersom Dockers standardpublicering av portar i bridge-nätverk kan kringgå UFW-regler — och gav driftsättningarna en oprivilegierad användare med ett enda fast kommando i stället för root.

Infrastruktur beskriven i ett kodarkiv

Byggde företagets egen infrastruktur som 19 Ansible-playbooks och 34 roller på 12 065 rader YAML, som konvergerar en värd i drift mot ett deklarerat tillstånd, där varje play är idempotent.

Ansvarade för plattformens Azure-driftsättningar

Ansvarade för plattformens driftsättningar till Azure från början till slut och hanterade releaser i utvecklings-, staging- och produktionsmiljöerna.

Provisionerade Azure-infrastruktur som kod

Provisionerade Azure-infrastruktur som kod med Bicep — Container Apps, PostgreSQL Flexible Server, Front Door/WAF och nätverk — i utvecklings-, staging- och produktionsmiljöerna.

Drev 20 Docker-appar på Google Cloud och AWS

Driftsatte och underhöll 20 containeriserade Docker-applikationer, felsökte med Podman och Kubernetes och förvaltade R-baserade appar på Google Cloud och AWS.

Tjänsten ingår i arbete som står annanstans på sidan:

Offentliga tjänster på en liten server · Byggde hela kärninfrastrukturen från noll · Förenklad infrastruktur utanför AWS

{ Moln, infrastruktur & DevOps }

Infrastruktur som kod

Servrar du kan bygga upp igen från versionshanterad konfiguration

Hela uppsättningen nedskriven som filer, så att den kan återskapas exakt och aldrig bara finns i en enda persons huvud.

Repeterade forgens CI-hook och hittade 2 fel

Repeterade CI-hooken på forge-sidan och hittade två fel som inte gick att se genom att läsa filen: en reservväg som lade ett argument som inte gick att lösa upp i hookens indata, och en miljövariabel från git själv som följde med grinden in i utcheckningen och gjorde 19 tester röda.

Delade upp varje driftshemlighet per värd

Delade upp de fyra driftshemligheterna per värd efter att ha konstaterat att två värdar som delar ett dödmansgrepp larmar mindre än två separata grepp, inte mer, och att en delad lösenfras för säkerhetskopior kan ge tillgång till båda värdarnas kopior, även i separata repon, om någon får tag på den och kan läsa båda reponas krypterade data och nyckelfiler.

Minskade systemd-exponeringen för varje enhet

Minskade exponeringen i systemd-sandlådan för varje enhet som plattformen installerar — en tjänst för dödmansgrepp från 9,6 UNSAFE till 1,5, en internetexponerad git-forge från 8,3 EXPOSED till 1,5 — och lade till en parserkontroll vid konvergering efter att ha hittat ett felstavat direktiv som tyst ignorerades i tre enhetsmallar.

Stämde av en DNS-zon med 20 poster som kod

Stämde av en DNS-zon med 20 poster deklarativt mot Cloudflare-API:t med separata ingångar för granskning och BIND-export, och stängde av CDN-proxyn igen av integritetsskäl efter att ha byggt den.

Kontrollerade hemligheter i förväg på en sekund

Lade till en play för förhandskontroll som kör samma kod som konvergeringen mot operatörens lokala hemligheter på ungefär en sekund, efter att en halvt tillämpad produktionskörning kraschade vid sin nionde uppgift med swapinställningarna redan skrivna till värden i drift.

Fick Ansibles check mode att tala sanning

Fick check mode att tala sanning i hela plattformen efter att ha hittat sex sonder som fattade beslut utifrån ett värde som värddatorn aldrig gav, eftersom command-modulen i Ansible rapporterar framgång under --check medan den hoppar över kommandot helt.

Byggde värdövervakning med dödmansgrepp

Byggde övervakning med dödmansgrepp som pingar bara så länge minne och disk är i gott skick, så att en försämrad värd larmar genom att tystna — och upptäckte sex variabelnamn som sa ”free” där kontrollen korrekt mätte ”available”, en storleksordning isär på en maskin med 464 MB.

Backupautomatik med en tydlig återläsningsmetod

Implementerade restic-backupautomatik med lagringspolicy, integritetskontroll och ett flöde för månatliga återläsningstest; dokumenterade aktivering av lagring och kontroll av en verklig återläsning som separata driftsteg.

Härdade SSH till 24 validerade direktiv

Härdade SSH till 24 kontrollerade direktiv med validering i tre steg — kandidatfilen, den sammansatta konfigurationen och sedan daemonens egen återläsning — efter att återläsningen hade avslöjat att den körande servern tyst åsidosatte två av de tjugofyra.

Tjänsten ingår i arbete som står annanstans på sidan:

Publicerade en läsbar onion-spegel i Tor · Provisionerade en andra server från kod · Er kod på er egen server · Offentliga tjänster på en liten server · Infrastruktur beskriven i ett kodarkiv · Byggde säkerhetskopior och återställning som kod · Provisionerade Azure-infrastruktur som kod · Skrev 1 000 RBAC-regler som säkrar Linux-servrar

{ Moln, infrastruktur & DevOps }

DevOps och CI/CD-automatisering

Leverera ändringar utan dramatik

Testade, släppta och möjliga att backa, så att ändringar når kunderna en tisdagseftermiddag i stället för sent en fredagskväll.

Lät 4 repon följa en och samma historikstandard

Lät fyra repon följa en och samma historikstandard — konventionella commits, fria från emojier, utan attributionsrader, upprätthållen av en hook för commit-meddelanden — vid sidan av 48 instruktionsdokument som styr hur arbetet utförs.

Tog testsviten ned till 8,9 sekunder

Tog testsviten från sex tester över en gräns på sextio sekunder till 135 godkända på 8,9 sekunder genom att profilera huvudtråden och ta bort de två anrop som den befann sig i vid 3 989 av 4 017 stickprov.

Upptäckte att hookarna och grinden var oense

Upptäckte att commit-hookarna och kvalitetsgrinden körde olika kontroller medan ett dokument lovade att de var desamma, genom att jämföra de två listorna i ett test — de fem som bara någonsin kördes för hand var just de som läste CV-texten.

Byggde en commit-grind med 22 linters

Byggde en commit-grind med 22 linters på en rad var plus fem som förtjänar ett eget stycke, utan varningsnivå och utan tillåtna inline-undantag, som täcker HTML, CSS, JavaScript, Python, YAML, Markdown, shell, länkar, stavning, hemligheter och typografi.

Drev ett löpande program för säkerhetsskanning

Etablerade ett kontinuerligt säkerhetsprogram — kodskanning, DAST, kontroller av beroenden och sårbarheter, generering av SBOM, skanning efter hemligheter och SHA-låsta actions — vid sidan av 21 skriftliga säkerhetsgranskningar.

Byggde repots kontrollmotor med 268 kontroller

Byggde repots kontrollmotor — 268 registrerade commit-kontroller, 277 lintregler och 15 egna ESLint-regler — plus 146 tester av själva kontrollerna, så att det är bygget som håller standarden, inte granskningen.

Automatiserade teamets samarbete

Automatiserade teamets samarbete, lösenordshantering samt uppgifts- och tidshantering och byggde ett halvautomatiskt system för att visa upp projekt, vilket höjde teamets produktivitet.

Satte ribban vid noll varningar i sex språk

Satte kvalitetsribban vid noll varningar i sex språk — Go, TypeScript, SQL, Python, Shell och Markdown — upprätthållen av pre-commit-hookar.

Skrev 578 automationsmål i go-task

Skrev 578 automationsmål i go-task som täcker utvecklingslägena native, Docker och HTTPS samt lintning, testning, databas och driftsättning.

Byggde CI/CD-pipelines i GitHub Actions

Byggde CI/CD-pipelines i GitHub Actions med en distroless-image för frontenden i produktion och uppflyttning mellan flera miljöer.

Effektiviserade CI/CD och sparade 4 000 timmar

Effektiviserade CI/CD-processerna och sparade 4 000 timmar genom att införa automatisering i pipelines för mjukvaruutveckling.

Automatiserade SSL/TLS-cert för 100 Docker-appar

Automatiserade skapandet av SSL/TLS-certifikat för 100 Docker-applikationer och säkerställde säkra anslutningar på alla värdar med Ubuntu Linux.

Minskade driftrisken med Grafana och Prometheus

Minskade driftriskerna genom att införa en dashboard för övervakning med Grafana och Prometheus, vilket förbättrade systemets tillförlitlighet.

Tjänsten ingår i arbete som står annanstans på sidan:

Känn till licensen för allt ni levererar · Tog i bruk den oanvända halvan av kärnan · Skrev en parser som verifierar varje FFI-anrop · Repeterade forgens CI-hook och hittade 2 fel · Skrev tester för själva kontrollerna · Aktiverade varje lintregel på felnivå · Gav generatorn 981 tester och 92 % täckning · Lintade själva kvalitetsgrinden på 10 242 rader · Rättade en sitemap med en enda delad tidsstämpel · Speglade webbplatsen till Gemini och Gopher · Ett skyddsnät som inte längre kostar en halvtimme · Körde containrar på Podman och Quadlet · Er kod på er egen server · Kontrollerade hemligheter i förväg på en sekund · Fick Ansibles check mode att tala sanning · Infrastruktur beskriven i ett kodarkiv · 1 022 migreringar bakom en grind mot schemadrift · Byggde egen felövervakning och spårning · Byggde en testsvit för Go, SQL och webbläsare · Byggde hela kärninfrastrukturen från noll · Ansvarade för plattformens Azure-driftsättningar · Genererade API-kontraktet utåt från databasen · Förenklad infrastruktur utanför AWS · Automatiserade databearbetning i Python, PL/pgSQL · Sparade 4 000 timmar med GitHub och GitLab CI/CD · Förnyade interna processer, sparade 8 000 timmar · Drev 20 Docker-appar på Google Cloud och AWS · Automatiserade 100 kritiska säkerhetskopior · Automatiserade leveransen av 20 GIS-datapipelines · Automatiserade GIS-driftsättning i GitHub Actions

{ Moln, infrastruktur & DevOps }

Containerisering och orkestrering

Mjukvara som fungerar likadant överallt

Paketerad för att bete sig identiskt på en laptop, på en testserver och i produktion, och för att skala upp när en hektisk vecka kommer.

Tjänsten ingår i arbete som står annanstans på sidan:

Körde containrar på Podman och Quadlet · Minskade systemd-exponeringen för varje enhet · Byggde CI/CD-pipelines i GitHub Actions · Automatiserade SSL/TLS-cert för 100 Docker-appar · Drev 20 Docker-appar på Google Cloud och AWS

{ Moln, infrastruktur & DevOps }

Tillförlitlig drift och övervakning

Vet före dina kunder

Övervakning och larm som säger till medan ett fel fortfarande är litet, och siffrorna som visar hur ofta det förblir så.

Fick sex insamlare att ange vad de hoppade över

Fick sex rapporterande insamlare att ange den del av sin indata som de aldrig läste — filer som förblev stängda, sonder som inte installerats, journalrader som inte kändes igen — och anpassade tröskelvärden till 116 mätningar, efter att ha konstaterat att en blind insamling och en värd i gott skick ger samma sida.

Byggde 7 skrivskyddade rapportroller för värdar

Byggde sju skrivskyddade rapportroller som återger en värd i drift som Markdown — fakta, åtkomst, git, mätvärden, trafik, säkerhet och leverantörsinventering — under regeln att inget tal skrivs ut som körningen inte har mätt.

Bevisade att intrångsspärren fungerar varje gång

Bevisade spärrkedjan mot intrång från början till slut vid varje härdningskörning genom att spärra en reserverad testadress, läsa den kärnregel som uppstod och häva spärren i ett städblock som alltid körs, så att en jail som slutar fungera fäller körningen i stället för att rapportera att allt är i ordning.

Tre lås på dörren, inget av dem låst

Hittade och åtgärdade tre skydd mot brute force-attacker över SSH som aldrig hade fungerat: en jail som bevakade port 22 medan daemonen lyssnade på 1986, en hastighetsbegränsning som skuggades av en bredare regel ovanför den och en spärråtgärd vars binärfil aldrig gick att hitta, så att ingen spärr någonsin hade tillämpats.

Höll strömmade IPTV-signaler igång oavbrutet

Säkerställde oavbruten leverans av strömmade IPTV-signaler mellan leverantörer och kunder och övervakade och underhöll strömningsnätet och IP-telefonin dygnet runt.

Gav support 24/7 för ca 1 000 IPTV/OTT-servrar

Gav infrastruktursupport dygnet runt, 24/7, för en IPTV/OTT-strömningsplattform och administrerade ca 1 000 servrar samt kundägda system för globala kunder i Kina, USA och Tyskland.

Tjänsten ingår i arbete som står annanstans på sidan:

Mätte 13 adresser helt utan spårning · Ett begränsat händelseflöde i en macOS-app · Delade upp varje driftshemlighet per värd · Fick Ansibles check mode att tala sanning · Byggde värdövervakning med dödmansgrepp · Backupautomatik med en tydlig återläsningsmetod · Byggde hastighetsbegränsning som stänger vid fel · Byggde egen felövervakning och spårning · Flyttade långsamt arbete till en jobbkö i River · Byggde infrastruktur som ännu körs år senare · Ansvarade för plattformens Azure-driftsättningar · Byggde e-post med failover och leveransloggning · En kartprodukt, buren hela vägen · Administrerade 40 webbplatser på Apache och Nginx ·

Minskade driftrisken med Grafana och Prometheus ·
Drev 10 PostgreSQL- och MS SQL-servrar på Ubuntu ·
Byggde företagets första observerbarhetsdashboard

{ Moln, infrastruktur & DevOps }

Säkerhetskopiering och återställning

Få tillbaka allt efter en dålig dag

Säkerhetskopior som testas i stället för att tas för givna, och en skriftlig plan för den morgon då en server inte startar igen.

Förvaltade 30 VPS-instanser med Ubuntu Linux

Förvaltade 30 VPS-instanser med Ubuntu Linux, införde strategier för katastrofåterställning och säkerställde optimala nätverkskonfigurationer.

Tjänsten ingår i arbete som står annanstans på sidan:

Delade upp varje driftshemlighet per värd · Backupautomatik med en tydlig återläsningsmetod ·
Byggde säkerhetskopior och återställning som kod · Automatiserade 100 kritiska säkerhetskopior

{ Moln, infrastruktur & DevOps }

Systemadministration

Håller allt igång

Daglig skötsel av servrar, e-post, konton och uppdateringar, så att det som alla förlitar sig på förblir odramatiskt.

Verifierade brandväggsregler efter position

Verifierade brandväggsregler efter position snarare än förekomst och läste den numrerade regellistan och den aktiva paketfilterkedjan, eftersom en regel som finns inte är detsamma som en regel som något paket når.

Införde en företagsgemensam lösenordshanterare

Integrerade ett företagsövergripande system för lösenordshantering, vilket stärkte säkerheten och effektiviserade åtkomstkontrollen.

Administrerade 100 Bare Bone-servrar

Administrerade 100 Bare Bone-servrar, fysiska nätverk och system för IP-telefoni och säkerställde en robust infrastruktur för företagets tillväxt.

Konfigurerade och driftsatte 1 000 Wi-Fi-routrar

Konfigurerade och driftsatte 1 000 Wi-Fi-routrar och förbättrade nätverkets tillgänglighet och prestanda för kunderna.

Administrerade nätverksinfrastrukturen

Administrerade nätverksinfrastrukturen för över 1 000 servrar och säkerställde optimal driftsättning, säkerhet och felsökning av systemen.

Förvaltade och felsökte 8 VPN-anslutningar

Förvaltade och felsökte 8 VPN-anslutningar över WireGuard och IPsec och säkerställde säker kommunikation i Google Cloud och på Linux-system.

Ledde behörighetshanteringen i M365 och SSO

Förhindrade säkerhetsintrång genom att leda initiativ för behörighetshantering med M365, 1Password, Red Hat SSO och OKTA SSO.

Tjänsten ingår i arbete som står annanstans på sidan:

Publicerade en läsbar onion-spegel i Tor · Fick sex insamlare att ange vad de hoppade över · Repeterade forgens CI-hook och hittade 2 fel · Er kod på er egen server · Byggde 7 skrivskyddade rapportroller för värdar · Minskade systemd-exponeringen för varje enhet · Stämde av en DNS-zon med 20 poster som kod · Byggde värdövervakning med dödmansgrepp · Backupautomatik med en tydlig återläsningsmetod · Bevisade att intrångsspärren fungerar varje gång · Härdade SSH till 24 validerade direktiv · Offentliga tjänster på en liten server · Infrastruktur beskriven i ett kodarkiv · Byggde hela kärninfrastrukturen från noll · Byggde infrastruktur som ännu körs år senare · Höll strömmade IPTV-signaler igång oavbrutet · Gav support 24/7 för ca 1 000 IPTV/OTT-servrar · Administrerade 40 webbplatser på Apache och Nginx · Förvaltade 30 VPS-instanser med Ubuntu Linux · Skrev 1 000 RBAC-regler som säkrar Linux-servrar · Drev 10 PostgreSQL- och MS SQL-servrar på Ubuntu

{ Moln, infrastruktur & DevOps }

Nätverk och VPN-uppsättning

Kopplar ihop dina kontor och dina medarbetare

Wi-Fi, brandväggar och fjärråtkomst uppsatta så att personalen kan arbeta varifrån som helst och bara rätt personer kommer in.

Tjänsten ingår i arbete som står annanstans på sidan:

Provisionerade en andra server från kod · Stämde av en DNS-zon med 20 poster som kod · Verifierade brandväggsregler efter position · Tre lås på dörren, inget av dem låst · Höll strömmade IPTV-signaler igång oavbrutet · Provisionerade Azure-infrastruktur som kod · Administrerade 100 Bare Bone-servrar · Konfigurerade och driftsatte 1 000 Wi-Fi-routrar · Administrerade nätverksinfrastrukturen · Förvaltade och felsökte 8 VPN-anslutningar · Förvaltade 30 VPS-instanser med Ubuntu Linux

{ Moln, infrastruktur & DevOps }

Säkerhet och behörighetshantering

Bestämmer vem som når vad

Konton, lösenord och behörigheter ordnade så att alla har exakt den åtkomst deras arbete kräver, och de som slutar förlorar den samma dag.

Förbättrade teamets kommunikation och samarbete

Förbättrade teamets kommunikation och samarbete genom att införa Slack, Mattermost, 1Password och Jira, vilket sparade 8 000 arbetstimmar.

Tjänsten ingår i arbete som står annanstans på sidan:

Publicerade en läsbar onion-spegel i Tor · Mätte 13 adresser helt utan spårning · Känn till licensen för allt ni levererar · Tog i bruk den oanvända halvan av kärnan · Delade upp varje driftshemlighet per värd · Skrev en avgränsningsregel efter läckta uppgifter · Provisionerade en andra server från kod · Körde containrar på Podman och Quadlet · Minskade systemd-exponeringen för varje enhet · Stämde av en DNS-zon med 20 poster som kod · Kontrollerade hemligheter i förväg på en sekund · Verifierade brandväggsregler efter position · Bevisade att intrångsspärren fungerar varje gång · Härdade SSH till 24 validerade direktiv ·

Tre lås på dörren, inget av dem låst · Drev ett löpande program för säkerhetsskanning · Byggde anspråk på företagsägarskap och prövning · Byggde hastighetsbegränsning som stänger vid fel · Byggde åtkomsträttigheter via Stripe och appköp · Införde en företagsgemensam lösenordshanterare · Automatiserade teamets samarbete · Härdade applikationen med nonce-baserad CSP · Provisionerade Azure-infrastruktur som kod · Byggde e-post med failover och leveransloggning · Byggde växling av organisationskontext · Automatiserade SSL/TLS-cert för 100 Docker-appar · Administrerade nätverksinfrastrukturen · Förvaltade och felsökte 8 VPN-anslutningar · Ledde behörighetshanteringen i M365 och SSO · Skrev 1 000 RBAC-regler som säkrar Linux-servrar

{ Moln, infrastruktur & DevOps }

Säkerhet för AI-agenter

Hålla AI:n borta från det den inte ska röra

Varje agents behörigheter kartlagda och begränsade till det nödvändiga, varje handling loggad där agenten inte kan ändra loggen, och varje skydd prövat i stället för taget för givet.

Tjänsten ingår i arbete som står annanstans på sidan:

Delade upp varje driftshemlighet per värd · Skrev en avgränsningsregel efter läckta uppgifter · Kontrollerade hemligheter i förväg på en sekund · Bevisade att intrångsspärren fungerar varje gång · Tre lås på dörren, inget av dem låst · Drev ett löpande program för säkerhetsskanning · Byggde hastighetsbegränsning som stänger vid fel · Ledde behörighetshanteringen i M365 och SSO · Skrev 1 000 RBAC-regler som säkrar Linux-servrar

{ Rådgivning & ledarskap }

Tekniskt ledarskap och rådgivning

En senior ingenjör i rummet

Någon på din sida av bordet när en leverantör, en budget eller ett arkitekturbeslut behöver en andra välgrundad åsikt.

Satte dokumentationen under en egen linter

Satte dokumentationen under en egen linter — en radbudget per fil som bara kan skruvas nedåt, ett krav på index och en kontroll av verkliga sökvägar — som hittade sex av nio planer markerade som klara trots okryssade rutor, och fjorton commit-spärrande uppgifter som inget dokument nämnde.

Vägledde företagets projekt genom utvecklingen

Vägledde genomförandet av ett stort antal projekt i företaget och gav expertstöd i mjukvaruutvecklingens olika faser.

Ledde tillväxten från 4 till 14 anställda

Ledde företagets tillväxt från 4 till 14 anställda med agila metoder och effektiv projektledning.

Skrev 50 000 ord programvarudokumentation

Skrev 50 000 ord utförlig programvarudokumentation i Markdown på GitHub, Craft och Confluence och säkerställde därmed att kunskapen bevarades och att processen var transparent.

Uppskattade 4 000 sparade timmar, team från 2–18

Uppskattade 4 000 sparade timmar genom handledning av ett team som växte från 2 till 18 medlemmar, bättre arbetsflöden och samordning mellan avdelningar.

Tjänsten ingår i arbete som står annanstans på sidan:

Lät 4 repon följa en och samma historikstandard · Tog testsviten ned till 8,9 sekunder · Införde strikt samtidighet i Swift 6 ovanpå C · Upptäckte att hookarna och grinden var oense · Skrev tester för själva kontrollerna · Aktiverade varje lintregel på felnivå · Gav generatorn 981 tester och 92 % täckning · Lintade själva kvalitetsgrinden på 10 242 rader · Drev utvecklingen som 51 skriftliga initiativ · Ett skyddsnät som inte längre kostar en halvtimme · Byggde repots kontrollmotor med 268 kontroller · Byggde en testsvit för Go, SQL och webbläsare · Levererade analyser och lägesrapporter till vd:n · Fattade plattformens tekniska grundbeslut · Satte ribban vid noll varningar i sex språk · Effektiviserade CI/CD och sparade 4 000 timmar · Förnyade interna processer, sparade 8 000 timmar · Ledde över 30 GIS-projekt från bygge till support · Styrde full stack-utvecklingen av GIS-kartor · Ledde GIS-kartappen till 10 gånger högre intäkter · En infrastrukturplan för 14 avdelningar

{ Rådgivning & ledarskap }

Teambygge och mentorskap

Utvecklar teamet du redan har

Rekrytering, onboarding och coachning, så att kunskapen stannar hos dina egna medarbetare i stället för att försvinna med en konsult.

Ledde ett team som levererade IT-support

Ledde ett team som levererade IT-support, dataräddning och hårdvarureparationer till över 1 000 kunder och säkerställde service av hög kvalitet.

Sparade 150 timmar per månad i 30 projekt

Effektiviserade projektarbetet och sparade 150 timmar per månad i 30 projekt genom att optimera arbetsflöden och resurshantering.

Etablerade en avdelning för teknisk support

Etablerade en avdelning för teknisk support som betjänade över 10 000 kunder med IT-support och felsökning och följde upp kundärenden i ett CRM-system.

Utformade och ledde 5 000 timmars kartutveckling

Utformade och ledde 5 000 timmars kartutveckling med Scrum och Jira för effektiv projektledning.

Tjänsten ingår i arbete som står annanstans på sidan:

Vägledde företagets projekt genom utvecklingen ·

Förbättrade teamets kommunikation och samarbete · Ledde tillväxten från 4 till 14 anställda ·

Ledde 30 framgångsrika GIS-projekt till leverans · Uppskattade 4 000 sparade timmar, team från 2–18

{ Rådgivning & ledarskap }

Projektledning (agil)

Får det levererat

Någon som ansvarar för omfattning, datum och de obekväma samtalen, så att ett projekt har en ägare i stället för fem åsikter.

Tjänsten ingår i arbete som står annanstans på sidan:

Lät 4 repon följa en och samma historikstandard · Satte dokumentationen under en egen linter · Drev utvecklingen som 51 skriftliga initiativ · Automatiserade teamets samarbete · Utformade ett tids- och rapporteringssystem · Levererade analyser och lägesrapporter till vd:n · Sparade 150 timmar per månad i 30 projekt · Etablerade en avdelning för teknisk support · Förnyade interna processer, sparade 8 000 timmar · Vägledde företagets projekt genom utvecklingen · Förbättrade teamets kommunikation och samarbete · Ledde tillväxten från 4 till 14 anställda · Ledde 30 framgångsrika GIS-projekt till leverans · Samlade in och analyserade verksamhetskrav · Utformade och ledde 5 000 timmars kartutveckling · Uppskattade 4 000 sparade timmar, team från 2–18

{ Rådgivning & ledarskap }

Produktstrategi och krav

Tar reda på vad som ska byggas

Omsätter det verksamheten vill ha till något en utvecklare kan bygga, och säger rakt ut vad som inte är värt att bygga.

En sammanhållen identitet i olika kanaler

Utvecklade Engineer ApS varumärkesidentitet i tryckt och digitalt material, med koppling till företagets ingenjörsportfölj. Grundaren beskriver en ökning av varumärkeskännedomen med 200 gånger som resultat av arbetet.

Tjänsten ingår i arbete som står annanstans på sidan:

Skrev om felmeddelanden som skyllde på användare · Drev utvecklingen som 51 skriftliga initiativ · Byggde en commit-grind med 22 linters · Byggde rekryterings- och karriärytor för sjöfolk · Byggde socialt skikt: inlägg, flöde och grupper · Byggde en lojalitetsplånbok, ligor och märken · Byggde anspråk på företagsägarskap och prövning · Byggde plattformens backoffice för operatörer · Byggde åtkomsträttigheter via Stripe och appköp · Utformade ett tids- och rapporteringssystem · Utformade hela produktens UI och UX · Utvecklade ett rapportsystem för dataanalys · Samlade in och analyserade verksamhetskrav · Utformade och ledde 5 000 timmars kartutveckling · Presenterade 200 UI/UX-förbättringar · Optimerade prognos- och investeringsstrategier · Ledde GIS-kartappen till 10 gånger högre intäkter · Varumärkeskommunikation utifrån publikens respons

{ Rådgivning & ledarskap }

Teknisk dokumentation

Skriver ner det så att nästa person kan följa

Överlämningar, driftinstruktioner och manualer som låter din egen personal driva systemet utan att ringa den som byggde det.

Tjänsten ingår i arbete som står annanstans på sidan:

Fick sex insamlare att ange vad de hoppade över · Lät 4 repon följa en och samma historikstandard · Skrev om felmeddelanden som skyllde på användare · Skrev en parser som verifierar varje FFI-anrop · Byggde en sifferkontroll mellan språken · Upptäckte att hookarna och grinden var oense · Klarade 106 av 106 PDF/UA-1-profilkontroller · Satte dokumentationen under en egen linter · Drev utvecklingen som 51 skriftliga initiativ · Byggde en commit-grind med 22 linters · Skrev en avgränsningsregel efter läckta uppgifter · Byggde 7 skrivskyddade rapportroller för värdar · Drev ett löpande program för säkerhetsskanning · Byggde repots kontrollmotor med 268 kontroller · Skrev 578 automationsmål i go-task · Migrerade API:t till Huma v2 med 100 % paritet · Skrev 50 000 ord programvarudokumentation · Levererade 8 Power BI-projekt med handböcker · En infrastrukturplan för 14 avdelningar

{ IT-stöd & hårdvara }

IT-support och helpdesk

Vardaglig datorhjälp

Laptopen, skrivaren, inloggningen och e-posten, fixade snabbt och på ett begripligt språk, utan att någon behöver känna sig dum.

Diagnostiserade och reparerade bärbara datorer

Diagnostiserade och reparerade hårdvara i bärbara datorer — skärmar, gångjärn, tangentbord, styrplattor, moderkort och strömförsörjning — och löste mjukvaruproblem i Linux, Windows och Mac.

Räddade data ur SD-kort, HDD:er, SSD:er och RAID

Utförde dataräddning från en lång rad medier — SD-kort, HDD:er, SSD:er, RAID-arrayer, externa diskar och Mac-system.

Hanterade 4 000 datorreparationer

Hanterade 4 000 datorreparationer och säkerställde snabb och effektiv lösning av hård- och mjukvaruproblem.

Tjänsten ingår i arbete som står annanstans på sidan:

Gav support 24/7 för ca 1 000 IPTV/OTT-servrar · Konfigurerade och driftsatte 1 000 Wi-Fi-routrar · Ledde ett team som levererade IT-support · Etablerade en avdelning för teknisk support

{ IT-stöd & hårdvara }

Hårdvarureparation och dataräddning

Räddar filer från en död maskin

Reparerar det som är värt att reparera och får ut fotografierna, fakturorna eller ritningarna från en disk som har slutat fungera.

Tjänsten ingår i arbete som står annanstans på sidan:

Diagnostiserade och reparerade bärbara datorer · Räddade data ur SD-kort, HDD:er, SSD:er och RAID · Hanterade 4 000 datorreparationer · Ledde ett team som levererade IT-support

{ Varumärke & marknadsföring }

Varumärke, marknadsföring och SEO

Bli hittad och ge rätt intryck

Ett namn, en profil och en webbplats som dyker upp när någon söker efter det du säljer, och som ger samma bild överallt där du syns.

Tjänsten ingår i arbete som står annanstans på sidan:

Byggde den visuella identiteten ur 2 teckningar · Byggde ett designsystem för macOS med 48 tokens · Rättade en sitemap med en enda delad tidsstämpel · Krympte stilmallspaketet från 87 KB till 31 KB · Ersatte 963 metadatablock med en enda graf · Stängde färgsystemet vid 28 färger ·

Genererade 495 sidor från en SQLite-export · Ersatte 27 teckenstorlekar med en skala i 6 steg · Kombinerade webbutveckling med SEO och innehåll · Varumärkeskommunikation utifrån publikens respons · En sammanhållen identitet i olika kanaler

Jämför ert utgångsläge

Välj ett jämförbart arbete och identifiera skillnaderna i er egen situation.

Se vad det kostar Hör av er →

Diskutera ett relevant angreppssätt

Be oss göra relevant erfarenhet till ett angreppssätt ni kan bedöma.

Se våra tjänster

welcome@engineer.company +45 50 10 61 84 Köpenhamn, Danmark

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/portfolio/