Vi bygger programvaren **teamet deres skal overta** — og skriver den slik at de kan.

Arkitektur, backender, frontender, nettsteder, kart, designsystemer og mer enn ett språk — fra København.

Engineer company
Programvare

Programvare bygd for å vedlikeholdes, ikke bare leveres

+45 50 10 61 84 · København, Danmark

Laget med kjærlighet av D.

Backender, frontender og arkitekturen som holder dem sammen, skrevet slik at den neste som åpner repositoriet, kan endre det. Arbeidsbeskrivelsene viser utgangspunktet, hva som ble gjort, og resultatet.

Begynn med én arbeidsflyt, integrasjon eller del av en applikasjon. Vi kan avtale en avgrenset leveranse med akseptansekriterier, dokumenterte grensesnitt og en overlevering teamet kan arbeide videre med.

Programvarearbeid · Priser og samarbeidsformer · Beskriv oppdraget

Begynn med resultatet dere trenger

Fortell oss hva virksomheten deres skal oppnå. Vi knytter sammen programvare, data og infrastruktur rundt menneskene som bruker dem.

Se tjenestene våre Ta kontakt →

Se hvordan vi løser oppgavene

Finn tjenesten som passer målet deres, og se eksempler på hva som ble levert og hvordan.

Prosjekter Ta kontakt →

Les om arbeidet **i sin sammenheng** — beslutningene, leveransen og de beskrevne resultatene.

Se hele katalogen

Én partner for **11 tjenester** fra arkitekturen til den siste skjermen.

Bla gjennom katalogen

{ Programvareutvikling }

Bestemme hvordan helheten skal bygges

Plattform- og løsningsarkitektur

Formen på et system tegnet opp på papiret først, slik at delene passer sammen og kostnaden er kjent før noen skriver kode.

Tok i bruk streng samtidighet i Swift 6 over C

Tok i bruk fullstendig streng samtidighet i Swift 6 uten actors, og koblet en blokkerende C-hendelsesløkke til main actor gjennom én produsent, én konsument og én rekkefølge — etter å ha slått fast at én task per hendelse mister rekkefølgen grensesnittet er avhengig av.

En avgrenset hendelsesflyt i en macOS-app

Avgrenset hendelsesbuffere og logging etter en registrert macOS-hendelse med vekst på 41 MB i sekundet og 111 GB komprimerte sider på en maskin med 36 GB; en registrert full testkjøring reduserte kjernens logglinjer fra 610 996 til 1 411.

Bygde en native macOS-klient i Swift 6

Bygde en native meldingsklient for macOS i Swift 6 og SwiftUI — 11 141 linjer fordelt på 53 filer — over C-grensesnittet til en Rust-kjerne lenket inn som et statisk arkiv fra en låst revisjon.

Se alle 20 resultater

Slik arbeider vi

Forstå først

Vi setter oss inn i problemet før vi foreslår en løsning. Oppgaven dere kommer med, er starten på den samtalen, ikke slutten.

Bygg det for å vare

Holdbart framfor raskt, klarhet framfor finurligheter. Systemet må fortsatt kunne håndteres av dem som arver det.

Stå inne for det

Vi avtaler eventuell løpende drift eller støtte som del av oppdraget. Overleveringen gjør ansvar og videre kontaktvei tydelig.

{ Programvareutvikling }

Motoren bak appen eller nettstedet ditt

Backend- og API-utvikling

Delen brukerne aldri ser: hvor dataene ligger, hvilke regler som gjelder, og hvordan andre systemer får lov til å snakke med den.

Nådde den ubrukte halvdelen av kjernen

Tok i bruk den halvdelen av meldingskjernen applikasjonen aldri hadde brukt — overføring av sikkerhetskopier, forsvinnende meldinger, redigering og ny sending av meldinger, verifiserte invitasjoner, proxyer og krypteringspolicy — og kjørte hver test mot det ekte biblioteket uten mocks.

Skrev en parser som verifiserer hvert FFI-kall

Skrev en parser som leser den faktiske C-headeren på 7 308 linjer og verifiserer hvert kallsted, hver enum-konstant og at hver klasse som eier en peker, er final, etter at en håndskrevet plassholder-header lot kall til tre fjernede funksjoner kompilere, lenke og krasje.

Tok i bruk streng samtidighet i Swift 6 over C

Tok i bruk fullstendig streng samtidighet i Swift 6 uten actors, og koblet en blokkerende C-hendelsesløkke til main actor gjennom én produsent, én konsument og én rekkefølge — etter å ha slått fast at én task per hendelse mister rekkefølgen grensesnittet er avhengig av.

Se alle 32 resultater

{ Programvareutvikling }

Skjermbildene folk faktisk bruker

Frontend-utvikling

Grensesnitt som laster raskt, fungerer på mobilen og oppfører seg likt i alle nettlesere kundene dine måtte bruke.

Bygde et macOS-designsystem med 48 tokens

Bygde et designsystem med 48 tokens på plattformens eget glassmateriale og skrev linteren som avviser et magisk tall, en hardkodet skriftstørrelse eller en animasjon som ignorerer innstillingen for redusert bevegelse.

Ga alle 17 ikoner navn for skjermlesere

Ga hver kontroll med bare ikon i grensesnittet et navn for skjermlesere etter å ha funnet at send-knappen ble lest opp som «arrow up circle, button», og skrev linteren som krever etiketten innen åtte linjer fra ikonet.

En avgrenset hendelsesflyt i en macOS-app

Avgrenset hendelsesbuffere og logging etter en registrert macOS-hendelse med vekst på 41 MB i sekundet og 111 GB komprimerte sider på en maskin med 36 GB; en registrert full testkjøring reduserte kjernens logglinjer fra 610 996 til 1 411.

Se alle 30 resultater

{ Programvareutvikling }

Et fungerende produkt, fra ende til annen

Fullstack-produktutvikling

Ett team som tar en idé fra skisse til noe kundene dine kan logge inn på, i stedet for at den sendes videre mellom tre leverandører.

Bygde en native macOS-klient i Swift 6

Bygde en native meldingsklient for macOS i Swift 6 og SwiftUI — 11 141 linjer fordelt på 53 filer — over C-grensesnittet til en Rust-kjerne lenket inn som et statisk arkiv fra en låst revisjon.

Bygde en databasedrevet CV-generator

Bygde en databasedrevet generator for CV, referanser, portefølje og søknadsbrev i Python — 41 moduler, 10 580 linjer — som gjengir seks utdataformater fra én SQLite-kilde med 23 tabeller, satt sammen av en idempotent pipeline i 19 trinn.

Bygde et flerspråklig Hugo-nettsted på 3 språk

Bygde et flerspråklig statisk nettsted i Hugo med 108 malfiler, blant dem 53 partials, som publiserer hver side i fire representasjoner fra ett innholdstre på tre språk, og på nytt under sju fokuserte underdomener bygd fra det samme treet.

Se alle 16 resultater

{ Programvareutvikling }

Et nettsted du kan oppdatere selv

Utvikling av nettsteder og CMS

Et nettsted som er raskt, lett å finne og kan redigeres av dine egne ansatte, uten en regning hver gang en formulering endres.

Publiserte et lesbart onion-speil på Tor

Publiserte nettstedet som et Tor onion-speil på en lesbar adresse som begynner med engineer, med nøkkelen generert utenfor verten slik at den aldri havnet i dette repoet eller på en bærbar PC, og lot det være den ene av fire inngangsdører som aldri telles.

Bygde den visuelle identiteten fra 2 tegninger

Bygde selskapets ikon- og favicon-sett fra to håndlagde tegninger, med alle avledede filer regenerert av skript og en kontroll som feiler når en avledet fil ble committet før kilden.

Bestod 106 av 106 PDF/UA-1-profilkontroller

Dokumenterte PDF/UA-1-validering av ni dokumenter med 106 av 106 bestått kontroller i den konfigurerte profilen for universell utforming, mens arkivvarianten feilet på én av 146 regler.

Se alle 22 resultater

{ Programvareutvikling }

Synlig når kundene spør en AI

Klar for AI-agenter

Et nettsted eller en app som AI-assistentene kundene dine bruker, kan finne, lese og handle ut fra, kontrollert ved hver utgivelse i stedet for lovet én gang.

Drev utviklingen som 51 skriftlige initiativer

Drev utviklingen av nettstedet som 51 skriftlige initiativer fordelt på 54 plandokumenter, hvert med sine åpne avkrysningsbokser, sine målinger og beslutningene det avslo — blant dem seks av åtte funn om agentberedskap som ble avvist med begrunnelsen registrert.

Rettet et sitemap med ett felles tidsstempel

Rettet et sitemap der 172 av 176 URL-er delte ett endringstidspunkt, ved å hente datoen fra git-historikken etter å ha slått fast at eksporten skriver om hver fil ved hver kjøring.

Erstattet 963 metadatablokker med én graf

Byttet ut 963 anonyme blokker med strukturerte data, som gjentok selskapet 1 671 ganger, mot én lenket graf av 16 typer utledet fra stabile opprinnelsesidentifikatorer.

Se alle 5 resultater

{ Programvareutvikling }

La AI bruke programvaren du allerede har

Verktøy for AI-agenter og MCP-servere

Programvaren du allerede har, åpnet som verktøy en AI-agent kan kalle, uten at noe som allerede virker, må skrives om.

Bygde operatørenes backoffice for plattformen

Bygde back office for operatørene — rundt 40 administrasjonsruter og 128 komponenter for eierskapskrav, moderering, funksjonsflagg, cache og diagnostikk — slik at plattformen kan driftes uten databasetilgang.

Flyttet tregt arbeid over i en River-jobbkø

Flyttet tregt arbeid ut av forespørselsbanen og over i en River-jobbkø — 15 worker-moduler, 8 planlagte oppgaver og 20 pg_cron-jobber — slik at en forespørsel returnerer mens arbeidet bak den fortsetter.

Genererte API-kontrakten med databasen som kilde

Genererte API-kontrakten fra databasen og utover — OpenAPI, en typet TypeScript-klient på 44 076 linjer, 61 mock-handlere og grensene brukergrensesnittet håndhever — med en kontroll i hvert ledd som feiler ved avvik.

Se alle 5 resultater

{ Programvareutvikling }

Kart som svarer på spørsmål

GIS- og geodataløsninger

Alt som skjer på et sted — fartøy, kjøretøy, tomter, sensorer — lagt på et kart du kan søke i, måle på og dele.

Et kartprodukt, håndtert fra ende til annen

Utformet, utviklet og støttet infrastruktur, databehandling og en kartapplikasjon over 2 år, med både programvareleveranser og driftsvedlikehold.

Ledet 30+ GIS-prosjekter fra utvikling til støtte

Ledet utvikling, produksjonssetting og støtte for over 30 GIS-prosjekter, med ekspertise i teknologiene PostgreSQL, Bash, Python, JavaScript, GDAL, ArcGIS, PostGIS og Mapbox.

Ledet 30 vellykkede GIS-prosjekter til levering

Styrte gjennomføringen av 30 vellykkede GIS-prosjekter og viste lederskap i leveransen av innovative løsninger på tvers av fagfeltet.

Se alle 13 resultater

{ Programvareutvikling }

Gjøre bruken selvforklarende

UI/UX-design og designsystemer

Oppsett, ordvalg og flyt lagt opp slik at folk fullfører det de kom for å gjøre, i stedet for å ringe kundestøtten din.

Bygde den visuelle identiteten fra 2 tegninger

Bygde selskapets ikon- og favicon-sett fra to håndlagde tegninger, med alle avledede filer regenerert av skript og en kontroll som feiler når en avledet fil ble committet før kilden.

Skrev om feilmeldingene som skyldte på brukerne

Skrev om applikasjonens feilmeldinger i tråd med en skriftlig tonestandard etter at en avvist innlogging ga brukeren skylden for en skrivefeil når leverandøren egentlig krevde et appspesifikt passord, og dekket det med en test som navngir leverandøren.

Bygde et macOS-designsystem med 48 tokens

Bygde et designsystem med 48 tokens på plattformens eget glassmateriale og skrev linteren som avviser et magisk tall, en hardkodet skriftstørrelse eller en animasjon som ignorerer innstillingen for redusert bevegelse.

Se alle 20 resultater

{ Programvareutvikling }

På mer enn ett språk

Internasjonalisering og lokalisering

Programvare som leses naturlig i alle markeder du selger i, også datoene, valutaene og måten navn sorteres på.

Bygde en tallkontroll på tvers av språk

Bygde en innholdskontroll på tvers av språk som feiler når en oversettelse mister et tall den engelske teksten oppgir, og når et språk bruker en notasjon som ikke hører hjemme i det — og fant to danske beskrivelser som manglet en måleverdi, og seksten ukrainske tekstbiter med engelske anførselstegn.

Flyttet hver tekststreng inn i et innholdstre

Flyttet alle brukerrettede strenger ut av Python og inn i et innholdstre på 874 filer på tre språk, etter å ha funnet døde oversettelser ingen kunne se var døde, og en kontroll som i det stille vurderte en tredjedel av resultatene.

Bygde en databasedrevet CV-generator

Bygde en databasedrevet generator for CV, referanser, portefølje og søknadsbrev i Python — 41 moduler, 10 580 linjer — som gjengir seks utdataformater fra én SQLite-kilde med 23 tabeller, satt sammen av en idempotent pipeline i 19 trinn.

Se alle 8 resultater

{ Programvareutvikling }

En AI-assistent som kan faget ditt

Utvikling av skreddersydde AI-agenter

En agent koblet til systemene du allerede bruker, testet på saker dine egne fagfolk skriver, under oppsyn av folkene dine og aldri låst til én modell.

Fikk seks innsamlere til å si hva de hoppet over

Fikk seks rapporteringsinnsamlere til å oppgi den delen av inndataene de aldri leste — filer som forble lukket, sonder som ikke var installert, journallinjer som ikke ble gjenkjent — og tilpasset terskelverdier til 116 målinger, etter å ha slått fast at en blind innsamling og en vert der alt er i orden, gir den samme siden.

Skrev tester for selve kontrollverktøyene

Skrev tester for selve kontrollene etter å ha slått fast at en kontroll som bare får ren inndata, en dag vil melde rent fordi den ikke leste noe — plantet en stavefeil for å bekrefte at stavekontrollen finner den, og hentet et id-område fra databasen i stedet for fra et tall i testen.

Bygde en commit-port med 22 lintere

Bygde en commit-port med 22 enlinjes lintere pluss fem som fortjener et avsnitt, uten advarselsnivå og uten tillatte undertrykkinger i koden, som dekker HTML, CSS, JavaScript, Python, YAML, Markdown, shell, lenker, staving, hemmeligheter og typografi.

Se alle 5 resultater

Vi jobber med **plattformene** virksomheten deres allerede kjører på.







Apple Microsoft DigitalOcean AWS Google

Se alle teknologiene

Klar for AI-agentene kundene dine sender

Funnet

En robots-fil som sier hva agenter får lov til, et sitemap med sanne datoer, og hver endring meldt via IndexNow samme dag den går ut.

Lest

En ren Markdown-kopi av hver side, et llms.txt-kart for språkmodeller, og strukturerte data som sier hva dere gjør.

Brukbar

Tilgjengelige skjemaer og betjeningselementer samt et beskrevet API for programoperasjonene som er valgt for agenttilgang.

Se hvordan det virker Spør om nettstedet deres →

AI-agenter bygd for arbeidet i bransjen deres

Tilkoblet

Koblet til ERP, CRM, bookingsystem og innboks gjennom dokumenterte verktøy, hvert med én avgrenset oppgave.

Målt

Testet mot reelle saker ekspertene dine skriver, så hver endring i en prompt, et verktøy eller en modell prøves før den tas i bruk.

Under tilsyn

Hva den gjør alene og hva et menneske godkjenner, skrevet ned sammen med dere, håndhevet i verktøyene og sporet steg for steg.

Se hvordan agenter bygges Spør om bransjen deres →

Programvaren dere kjører, gjort om til verktøy agenter kan kalle

Avdekket

Reglene som skjuler seg i skjermbilder, skjemaer og lagrede prosedyrer: funnet, skrevet ned og samlet i ett lag.

Åpnet

Hver operasjon tilbudt som et typet, dokumentert verktøy via et API eller en MCP-server, og bare innenfor brukerens rettigheter.

I drift

En avgrenset integrasjon kan åpne én operasjon om gangen ved siden av eksisterende skjermbilder; endringer i applikasjonen avtales etter en gjennomgang av grensesnittene.

Se hvordan programvare åpnes Spør om programvaren deres →

Fortell oss hva dere holder på å bygge.

Ingen skjema, ingen trakt, ingen salgssekvens — bare én e-post til et menneske som leser den.

Start samtalen

Denne adressen viser bare programvarearbeidet. Selskapet arbeider også med data, drifter infrastruktur og rådgir team — den siden av det ligger på engineer.company.
Se alt Engineer ApS gjør →

Begynn med en avgrenset forbedring

En arbeidsflyt, en dataflyt eller et system som bør fungere bedre, kan være en nyttig første oppgave. Vi kan avtale omfanget sammen.

Se hva det koster Ta kontakt →

Forbered den første samtalen

Fortell oss hvilket resultat dere trenger, hva som finnes allerede, og når dere ønsker det klart. Da kan vi diskutere neste steg.

Se tjenestene våre

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

På serveren jobber vi med **Linux med åpen kildekode** — ingen lisenser å betale og ingenting skjult.

Ubuntu Debian

Slik drifter vi servere

Ta med deg denne siden Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://software.engineer.company/nb/