Engineer Mjukvara

{ 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.

Håller ni fortfarande på att reda ut vad ni behöver?

Berätta vad som inte fungerar eller vad ni försöker bygga. Vi säger vad det skulle kräva, ungefär vad det skulle kosta och om vi är rätt personer för uppgiften.

Från 1 400 DKK i timmen, exklusive moms — och mindre per timme vid en längre bokning.

Hör av er Se vad det kostar

Utvalt arbete som visar den här tjänsten.

Gör tjänsten till ett resultat

Låt ”Backend- och API-utveckling” stödja ett affärsmål. Vi kan knyta det tekniska arbetet till användarna, driften och besluten det ska stödja. Den del användarna aldrig ser: där data lagras, vilka reglerna är och hur andra system får kommunicera med den.

Hör av er

Granska det relevanta arbetet

Använd arbetet på sidan för att granska vårt angreppssätt. Fråga om avvägningarna och förutsättningarna bakom varje resultat. Utformade och drev 6 ETL/ELT-pipelines — 6 ETL-pipelines över tre databasmotorer

Utformade och drev 6 ETL/ELT-pipelines Hör av er →

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.

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.

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.

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.

Utvecklade ett rapportsystem för dataanalys

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

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.

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.

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.

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.

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.

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.

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.

Implementerade en katalogstyrd djupsammanslagning

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

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.

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.

Byggde och underhöll Django-appar för IPTV

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

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 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.

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 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.

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 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 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 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 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.

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.

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.

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.

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.

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.

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.

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.

Letar ni efter Backend- och API-utveckling?

Använd de länkade arbetena för att bedöma relevant erfarenhet. Berätta vad ni behöver bygga; vi kan diskutera omfattning, arbetssätt och överlämning.

Hör av er Hela portföljen

Avgränsa uppdraget före beställning

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Gå igenom era krav med oss

Berätta vad ni behöver inom ”Backend- och API-utveckling”, vilken tidsram ni önskar och vilket team som ska förvalta lösningen. Vi kan diskutera en praktisk väg framåt.

Hör av er

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://software.engineer.company/sv/services/backend-development/