Engineer

Lösningsarkitektur

14 resultat

Koppla samman systemets delar

Arbetet under ”Lösningsarkitektur” visar en del av ett fungerande system. Vi kan hjälpa er att koppla den delen till resten av verksamheten. Systemets form utarbetad på papper först, så att delarna passar ihop och kostnaden är känd innan någon skriver kod.

Se våra tjänster Hör av er →

Hitta en jämförbar utmaning

Öppna ett exempel som liknar er situation. Genomgången förklarar besluten bakom resultatet, utöver teknikens namn. En infrastrukturplan för 14 avdelningar — 14 forskningsavdelningar under en och samma infrastrukturplan

En infrastrukturplan för 14 avdelningar Hör av er →

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.

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.

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.

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 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.

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 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.

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.

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.

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.

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.

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.

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.

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.

Fokusera den tekniska insatsen

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 →

Sätt ämnet i ert sammanhang

Berätta hur ”Lösningsarkitektur” ingår i ert system. Vi kan hjälpa er att välja rätt tjänst och avgränsa arbetet.

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/categories/solution-architecture/