



Backend-utvikler

Data Engineer

Peter Bull Hove

Utvikler

Peter er en utvikler med ekspertise innen C#/.NET og Data Platform. Han brenner for gode team og en effektiv arbeidshverdag. Peter studerte kybernetikk og robotikk ved NTNU, med fordypning i autonome systemer. Han er engasjert, og har vært både team-lead og scrum master i team hvor han har jobbet for Miljødirektoratet og Equinor.

Peter syntes det var spennende å bygge opp en RDF-database i C#/.NET for Equinor, og han syntes det var kult å designe arkitekturen til og implementere automatiske datapielines med Databricks og SQL for SFI Smart Ocean.

Han er glad i å dele kunnskap, og Peter har holdt flere presentasjoner på tech-konferanser, for eksempel TDC i Trondheim og Booster-konferansen i Bergen. På fritiden liker han å kjøre ski i bratte fjellsider og å spille volleyball. Peter har og en lidenskap for teater. Han har startet opp, ledet og spilt i to teaterstykker.

Språk/plattform

C#, RDF, JSON, .NET, TypeScript, Python, .NET Core, JavaScript, Sparql, Parquet

Rammeverk

Entity Framework Core, SignalR, PySpark, Numpy, React, React Query, PyTest, Pandas, Prettier, FastAPI, React Hook Form, Tanstack Query, Delta Lake, Recharts, Tailwind CSS, Koalas

Skyteknologi

Microsoft Azure, Azure Application Insights, Azure Web App, Azure Key Vault, Azure Blob Storage, Azure Databricks, Azure SQL Database, Azure Functions, Azure Event Hubs, Azure Static Web Apps

Database

SQL, Azure CosmosDB, MongoDB, NoSQL, Power BI

Bransjer

Olje og gass • Havbruk • Offentlig

Prosjekterfaring

04.2024 - 08.2024

Sett- og Skutt App for Miljødirektoratet

"Sett og Skutt" er en app for jegere som driver med hjorteviltjakt i Norge, med over 50.000 aktive brukere. Appen lar jegere registrere dager de har jaktet, antall dyr de har skutt, samt detaljer om dyrene. Denne informasjonen er påkrevd å sende inn til Miljødirektoratet, og "Sett og Skutt" tilbyr en enkel og brukervennlig måte å gjøre dette på.

De fleste jegere bruker appen ute i felt, derfor er det nødvendig at den fungerer godt i områder med dårlig eller ingen dekning. Appen bruker GPS for å stedfeste posisjonen til jegeren og det felte dyret. Den gir også mulighet for å registrere jaktlag med flere personer, hvor hvert medlem kan fylle ut informasjon på sine mobile enheter. Appen sammenstiller og validerer denne informasjonen, noe som sikrer nøyaktighet og konsistens.

Prosjektet innebar en total omskriving av appen og lansering av en ny versjon. Den gamle appen var skrevet på utdatert teknologi og oppfylte ikke lengre kravene til sikkerhet, oppetid og vedlikeholdbarhet. Den nye appen er en Progressive Web Application (PWA) skrevet i React og TypeScript, og kommuniserer med et REST API utviklet i C#/.NET.

Rolle: Utvikler

Peter jobbet som React/Typescript-utvikler i teamet som utviklet den nye versjonen av "Sett og Skutt". Han utviklet skjema, komponenter, sider og menyer som brukerne interagerer med når de bruker appen. Han hadde fokus på at appen skal være enkel, rask og brukervennlig. I tillegg til dette skrev han CI/CD pipelines i Github Actions for testing og utrulling av kode til test- og produksjonsmiljø, og fikset noen bugs i backenden som var et REST-API skrevet i C#/.NET.

Peter fulgte opp automatiske sikkerhetsanalyser av kodebasen, og passet på at appen ikke hadde noen sikkerhetshull eller sårbarheter. Han videreutviklet autentiseringen for appen, med OAuth2.0 og jobbet med henting, lagring og validering av JWT-tokens.

Han var også med på å utvikle caching for appen ved å bruke React Query, slik at appen skal fungere med lite eller manglende nettilgang. Ved å bruke en service worker implementerte Peter funksjonen at appen oppdateres automatisk på brukernes enheter, samt at den pre-cacher deler av appen. Peter gjorde også analyser av responstiden til appen, og komprimerte mediefiler og fjernet kodepakker, slik at appen ble raskere.

Han tok en sentral rolle i prosjektet, og samarbeidet tett med kunden og de andre utviklerne på teamet. I løpet av prosjektet etterspurte han ofte brukertester, tilbakemeldinger på arbeidet som var gjort.

Anvendt Kompetanse:

React, TypeScript, JavaScript, C#, .NET, Git, GitHub, GitHub Actions, Unit Testing, CI/CD, OAuth2.0, Prettier, Biome, Jest, Microsoft Azure, Azure Static Web Apps, REST API, SQL, GitHub Copilot, JSON Web Tokens (JWT), Azure Application Insights, JSON, Kanban, Smidig, Tanstack Query, React Query, React Hook Form, Visual Studio Code, Atlassian Jira, Continuous Deployment, Continuous Integration, Jira, Tailwind CSS

04.2024 - 04.2024

Lost Circulation Material for Equinor (Statoil ASA)

Lost Circulation Material (LCM) er en webapplikasjon designet for å beregne de beste blandeforholdene av materialer som brukes i petroleumdrilling.

Ved petroleumdrilling oppstår ofte mikrosprekker i fjellet rundt borehullet. For å forsegle disse sprekkenes og bevare fjellets styrke, benyttes en spesialblandet væske som består av forskjellige materialer. Valget av materialer i

borevæsken er avhengig av mange faktorer, og å finne den optimale blandingen er en kompleks oppgave.

LCM-applikasjonen består av et dashboard der brukerne kan velge innstillinger og parametere, samt en optimaliserer som beregner den beste blandingen av materialer. Optimalisereren benytter en genetisk algoritme som genererer tilfeldige blandinger og simulerer naturlig evolusjon, basert på hvilke sammensetninger som gir de beste resultatene.

Dashbordet inkluderer også flere grafer og verktøy for sammenligning av ulike blandinger, noe som gjør det enklere for brukerne å finne og implementere den mest effektive løsningen for deres spesifikke behov.

Rolle: **Fullstack-utvikler**

Peter jobbet som fullstack-utvikler med Python backend og React/Typescript frontend. Backend var et REST-API laget med Python FastAPI. Han skrev om deler av API'et for å tilgjengeliggjøre mer data for brukeren, samt å refaktorere koden til et enklere system.

Peter skrev og designet nye grafer i web-appen for visualisering av data ved å bruke Recharts, samt redesignet han dashboardet for å skape en bedre og enklere flyt for brukerne.

Peter skrev også om mye av appen fra Javascript til Typescript, og brukte Docker og Github Actions for å utrullere koden til Azure Web Apps til både test og produksjonsmiljø.

Anvendt **Kompetanse**:

Python, FastAPI, React, TypeScript, CI/CD, Git, Github Actions, Github, Unit Testing, PyTest, REST API, Docker, Microsoft Azure, Azure Web App, Genetic Algorithm, Docker-compose, Azure Web Services, JavaScript, Prettier, JSON, Visual Studio Code, Github Boards, Recharts

06.2023 - 04.2024

Marine Analysis Framework for Equinor (Statoil ASA)

Sikkerheten til marine systemer er sterkt avhengig av nøyaktige simuleringer og oppdaterte analyser. Operatører trenger kontinuerlig oppdaterte modeller og enkel tilgang til analyser for å kunne håndtere marine installasjoner på en sikker og effektiv måte.

Utviklingen av analyse- og simuleringsverktøy kan være tidkrevende, selv for mindre verktøy. Mange av disse verktøyene har lignende strukturer og kan bygges ved hjelp av samme grunnleggende komponenter. Derfor ble det utviklet et lavkode-rammeverk kalt Marine Analysis Framework (MAF). MAF integrerer data- og prosessmodellering, datalagring, frontend, og analyseorkestrering i et samlet rammeverk, som gjør det mulig for brukerne å raskt opprette analyseverktøy som webapplikasjoner.

MAF standardiserer og forenkler utviklingen av applikasjoner og analyseverktøy. Operatører og domeneeksperter innen marine analyser kan beskrive datamodeller og kravspesifikasjoner ved hjelp av høynivå skjemaer, og resten av prosessen håndteres automatisk av MAF.

Dette rammeverket gjør det lettere for operatører å få tilgang til nøyaktige og pålitelige analyser, og bidrar til en sikrere og mer effektiv forvaltning av marine systemer.

Rolle: **Fullstack-utvikler**

Peter utviklet og vedlikeholdt et REST-API i Python. Han brukte Docker for å deploye API'et sammen med en MongoDB database. Endepunktene i API'et håndterte store datamengder, validerte data automatisk og lagret data og binærfiler på en enkel og strukturert måte. Han skrev han også mange enhetstester og intergrasjonstester for API'et.

Peter designet og implementerte visuelle komponenter for rammeverket i React/Typescript. Peter tok ansvar for redesign av mye av det visuelle i MAF, med fokus på god brukervennlighet og å skape et godt og konsistent visuelt utseende. Peter videreutviklet bruk av caching for å optimalisere kjøretid på MAF. Dette gjorde at analyseverktøyene ble mer raskere og mer brukervennlige.

I tillegg bestod arbeidet i vedlikehold av utrullingsmiljø i Microsoft Azure og Equinor sin skyløsning Radix. Peter skrev automatiske CI/CD pipelines i Github Actions for å teste og utrullere kode automatisk, samt publisere nyeste versjoner av MAF som npm-pakker. Han tok ansvar for å starte en bedre tilbakemeldingskultur i teamet, og arrangerte retrospektiver og andre workshops som forbedret effektiviteten i teamet.

Anvendt **Kompetanse:**

Python, FastAPI, React, TypeScript, JavaScript, Docker, Docker-compose, Prettier, JSON Schema, JSON, Unit Testing, Integrasjonstest, Playwright, Azure Web App, MongoDB, NoSQL, Biome, REST API, Kanban, GitHub, GitHub Actions, Git, CI/CD, Azure CosmosDB, Azure Key Vault, Azure Blob Storage, Continuous Delivery, npm, Visual Studio Code, Continuous Deployment, Continuous Integration, REST, BDD, snyk, Tailwind CSS

01.2022 - 02.2023

SSI - Database og API for å muliggjøre Industri 4.0 for Equinor (Statoil ASA)

Strukturering av informasjon gjør det mulig å samle data fra forskjellige kilder, og gjøre analyser og validering på tvers av kildene. Spine Semantic Infrastructure (SSI) er en samling av databaser og API for å håndtere semantisk strukturert data for Equinor. Dette ble anvendt på plattformen Krafla.

Ved å ha plattformdata på en semantisk strukturert måte kan det kjøres mer komplekse valideringer og analyser på dataen. I tillegg kan dataen brukes til å generere diverse dokumenter og skjema. Eksempelvis kan et design av en strømkrets lagres som en semantisk strukturert graf. Da kan det kjøres en validering på strømkretsen for å se om det er en gyldig krets, og et kretsskjema kan genereres ut i fra dataen.

I tillegg ble det implementert system for å logge versjoner av datapakker i SSI. Dermed er det enkelt å se loggen over revisjoner som er gjort på dataen, og det blir enklere å samhandle på tvers av applikasjoner som benytter seg av den samme dataen.

Rolle: **Backend-utvikler**

Peter spilte en sentral rolle i Industri 4.0-prosjektet, hvor han utviklet tre separate API-tjenester og mikrotjenester i C#/.NET. Disse API-ene var avgjørende for å hente inn strukturert informasjon og videreformidle den effektivt. API'ene håndterte filoplasting og streaming av store datafiler, validerte- og transformerte data. Peter designet også en event-drevet arkitektur med mikrotjenester å transformere og propagere data videre til andre tjenester. Han vedlikeholdt og utviklet SQL-databaser og binærfil-databaser i Azure med Azure Blob Storage. Peter synes det er viktig med gode CI/CD-pipelines, og det skrev han i Azure DevOps og Github Actions som automatiserte testing, utrulling av miljøer og deployment.

Rolle: **Scrum Master**

I tillegg til sin rolle som utvikler, fungerte Peter også som Scrum Master for teamet. Han hadde ansvar for å sikre progresjon og god flyt i teamet ved å fasilitere daglige standups, planleggingsmøter og kundemøter.

Anvendt **Kompetanse:**

C#, .NET, REST API, Microservices, Terraform, RDF, Resource Description Framework, Azure Web App, Azure Blob Storage, Azure SQL Database, Visual Studio Code, Prettier, Infrastructure as Code (IaC), Azure Event Hubs, OAuth2.0, Event Driven Architecture (EDA), JSON, Azure Functions, Azure DevOps, Azure DevOps Pipelines, Unit Testing, Integrasjonstest, Datavalidering, Azure Key Vault, snyk, Entity Framework Core

06.2021 - 12.2021

Dataplattform for analyse av havdata og fiskehelse for SFI Smart Ocean

SFI Smart Ocean er et konsortium bestående av flere havinteressenter som samarbeider om datainnsamling og dataanalyse for å fremme innovasjon og effektivitet i havnæringene. For å støtte dette samarbeidet utviklet ble det laget en avansert dataplattform for innsamling, transformasjon lagring og analyser av havdata.

Dataplattformen gjør data tilgjengelig for havinteressentene i SFI Smart Ocean og migrerer data til forskjellige bruksområder for videre analyse og anvendelse. Disse bruksområdene var blant annet optimalisering av foring og fiskehelse for oppdrettsnæringer og vedlikeholdsprediksjoner for energiselskaper basert på bølge- og værdata.

Ved å samle og analysere data på en strukturert og effektiv måte, bidrar dataplattformen til bedre beslutningsprosesser, økt innovasjon og bærekraftig utvikling for aktørene i SFI Smart Ocean.

Rolle: **Data Engineer**

Peter var teamleder for data engineering-teamet som utviklet dataplattformen. Han tok ansvar for teamets progresjon, hadde kontakt med kunden, og var teknisk ansvarlig.

Teamet og Peter utviklet dataplattformen på Microsoft Azure ved bruk av Python og C#/.NET. Peter skrev dataflyter og datavalideringsskript i Python på Databricks, som ble automatisk kjørt når ny data var tilgjengelig. Han vedlikeholdt en SQL-database i Azure med Entity Framework og lagde PowerBI-rapporter for å visualisere nøkkeldata til brukere.

Han utviklet og vedlikeholdt flere mikrotjenester og et REST-API i C#/.NET for innhenting og levering av data. Han arbeidet med å implementere og optimalisere skyarkitekturen til dataplattformen, ved bruk av Terraform.

Peter fokuserte på å sikre godt samarbeid i teamet, oppmuntre til læring og kunnskapsdeling, og bygge opp den nødvendige kompetansen i teamet. Dette gjorde han ved å fasilitere retrospektiver, sikre klarhet i oppgavene og hyppig kontakt med kunden.

Anvendt **Kompetanse:**

C#, .NET, Azure Databricks, Python, Data Engineering, Jupyter Notebook, Azure Web App, Azure Blob Storage, SQL, Azure SQL Database, Pandas, Numpy, Entity Framework, Sparql, PySpark, Delta Lake, Parquet, Medallion architecture, DataOps, Data scraping, Datavalidering, Datavask og -konvertering, PyCharm, Azure Key Vault, Terraform, Visual Studio Code, Git, GitHub, GitHub Actions, CI/CD, Continuous Deployment, Continuous Integration, Infrastructure as Code (IaC), Azure Functions, Entity Framework Core, REST API, Power BI, JSON, Koalas

Utdannelse

Norges teknisk-naturvitenskapelige universitet (NTNU)

Sivilingeniør, Kybernetikk og

Robotikk

Sertifiseringer

2024	Databricks Azure Platform Architect Databricks
2024	Databricks Generative AI Fundamentals Databricks
2024	Databricks Lakehouse Fundamentals Databricks
2024	Databricks Fundamentals Databricks
2024	SC-900: Security, Compliance, and Identity Fundamentals Microsoft
2024	DP-900: Azure Data Fundamentals Microsoft
2023	People Development ICP-PDV The International Consortium for Agile (ICAgile)
2023	ICP-LEA - Leading with Agility The International Consortium for Agile (ICAgile)
2023	OKR Certification: Leadership and Goal Setting Measure What Matters / Coursera
2022	Certified Scrum Master Agile Lean House AS
2021	AZ-900: Microsoft Azure Fundamentals Microsoft

Kurs

2024	Implementing Modern Architecture Bouvet Norge AS
2023	People Development - ICP-PDV ICAgile
2023	Leading with Agility - ICP-LEA ICAgile
2022	Booster Conference Booster
2022	Scrum Master-kurs Bouvet Norge AS
2021	Norwegian Developers Conference (NDC)

Kompetanse

Språk/plattform

C#, RDF, JSON, .NET, TypeScript, Python, .NET Core, JavaScript, Sparql, Parquet

Rammeverk

Entity Framework Core, SignalR, PySpark, Numpy, React, React Query, PyTest, Pandas, Prettier, FastAPI, React Hook Form, Tanstack Query, Delta Lake, Recharts, Tailwind CSS, Koalas

Skyteknologi

Microsoft Azure, Azure Application Insights, Azure Web App, Azure Key Vault, Azure Blob Storage, Azure Databricks, Azure SQL Database, Azure Functions, Azure Event Hubs, Azure Static Web Apps

Database

SQL, Azure CosmosDB, MongoDB, NoSQL, Power BI

Arbeidserfaring

2024 -	Miles AS	Utvikler, Konsulent
2021 - 2024	Bouvet ASA	Utvikler, Konsulent

Språk

Engelsk Flytende

Norsk Morsmål