

Student	Tittel	LinkedIn
Daniel Davik Møgster	Prosjektleder / Tech Lead	www.linkedin.com/in/danielmogster
Tobias Engedal Ek	Dokumentasjonsansvarlig / Utvikler	www.linkedin.com/in/tobias-engedal-ek-20288b28b
Oliver Lindberg Skundberg	Scrum Master / Utvikler	www.linkedin.com/in/oliver-lindberg-skundberg-b8202928a
Sigurd Munk Brekke	Utvikler	www.linkedin.com/in/sigurd-munk-brekke-059262251
Emil Stokken Kaasa	Utvikler	www.linkedin.com/in/emil-kaasa-752604291

Oppdragsgiver

Oppdragsgiveren vår har vært Norkart, et teknologiselskap som leverer løsninger innen kart og geodata. Kontaktperson har vært Alexander Salveson Nossun, som ga gruppen kontorplass og tett oppfølging gjennom prosjektløpet. Norkart gav gruppen en oppgave knyttet til hvordan KI og MCP kan benyttes som arbeidsverktøy for saksbehandlere som jobber med konsekvensutredninger og romlige analyser.

Hva prosjektet går ut på

Konsekvensutredninger er en lovpålagt del av norske plan- og utbyggingsprosesser, og kan strekke seg over hundrevis av sider med faglig innhold fordelt på regelverk, veiledere og geografiske datasett. Ifølge Norkart kan saksbehandlere bruke måneder på mindre prosjekter og over ett år på større. Mye av den tiden går til manuell informasjonsinnhenting fra kilder som ikke snakker sammen.

Atlas er vår løsning på dette. En fullstack webapplikasjon som kombinerer et interaktivt kart, en KI-basert chatbot og GeoMCP-servere for utførelse geografiske analyser. Her kan en saksbehandler arbeide i ett enkelt grensesnitt og utforske kartdata, stille spørsmål til en KI-assistent, og få analytiske svar tilbake som også kan lagres som visuelle kartlag i kartet. Resultatene kan eksporteres som blant annet GeoJSON eller PDF for videre bruk i saksbehandling.

Assistenten kan ta i bruk datasett fra en PostgreSQL database, hente dokumenter fra Azure Blob Storage, samt benytte MCP-verktøy for å utføre utregninger og besvare forespørsler fra en saksbehandler. Resultatet er et arbeidsbord der kart, dokumenter og fagdialog spiller sammen.

Utfordringer og kompleksitet

Den største tekniske utfordringen var å få KI-agenten til å oppføre seg forutsigbart, spesielt med tanke på når den skal slå opp noe i databasen, når den skal regne på et kart, når den skal lete i dokumenter, og når den bør holde igjen. Gruppen måtte bygge tydelige rammer rundt dette, slik at KI-agenten ikke kunne gjøre hva som helst.

Hukommelse i chatbot-funksjonen var en annen utfordring. Samtaler lagres i databasen, men selve AI-sesjonen lever i minnet på serveren. Hvis noe ryker eller brukeren blir borte en stund, var det nødvendig å kunne plukke opp tråden igjen uten at det føltes som et brudd. Det samme gjaldt strømmingen av svar, hvor det ser fint ut når det fungerer, men vi måtte tenke nøye gjennom hva som skjer dersom tilkoblingen faller ut midt i en setning.

Kart og geodata var heller ikke så enkelt som gruppen først tenkte, da koordinater kommer i ulike systemer, avstander skal måles korrekt, og geometrien må holde mål både teknisk og visuelt. Søket har også vært en utfordring, der vi blander vanlig tekstsøk, fuzzy-søk og semantisk søk basert på embeddings, og må leve med at ikke alt i datasettet er ferdig indeksert til enhver tid.

På den faglige siden var konsekvensutredning et ukjent felt alle gruppemedlemmer ved oppstart. Vi brukte første del av prosjektet til å sette oss grundig inn i Miljødirektoratets veileder M-1941, noe som i ettertid viste seg å være et for bredt omfang. Etter tilbakemelding fra veileder ved UiA og en ansatt ved Norkart, spisset gruppen fokuset mot saksbehandlerperspektivet og ett konkret KU-tema. Verdensarv ble valgt da det var mindre komplekst enn andre temaer, noe som ga gruppen en realistisk ramme å jobbe innenfor.

Til slutt har gruppen også opplevd flere brudd og mistet tilganger til KI underveis, noe som er en særdeles viktig del av et KI-drevet system. I februar varslet Norkart om at Azure-tilgangen mulig stod i fare, noe som ikke var så farlig da gruppen bevegde seg over til GitHub Copilot SDK arkitekturen. I mars derimot fjernet GitHub Anthropic-modellene gruppen benyttet som KI-agent i systemet fra studentabonnementet dagen før et planlagt demomøte. Gruppen har måttet omstille seg til nye KI-modeller og flere skifter i løpet av prosjektet, men heldigvis kom vi oss videre uten at det gikk ut over leveransen i nevneverdig grad.

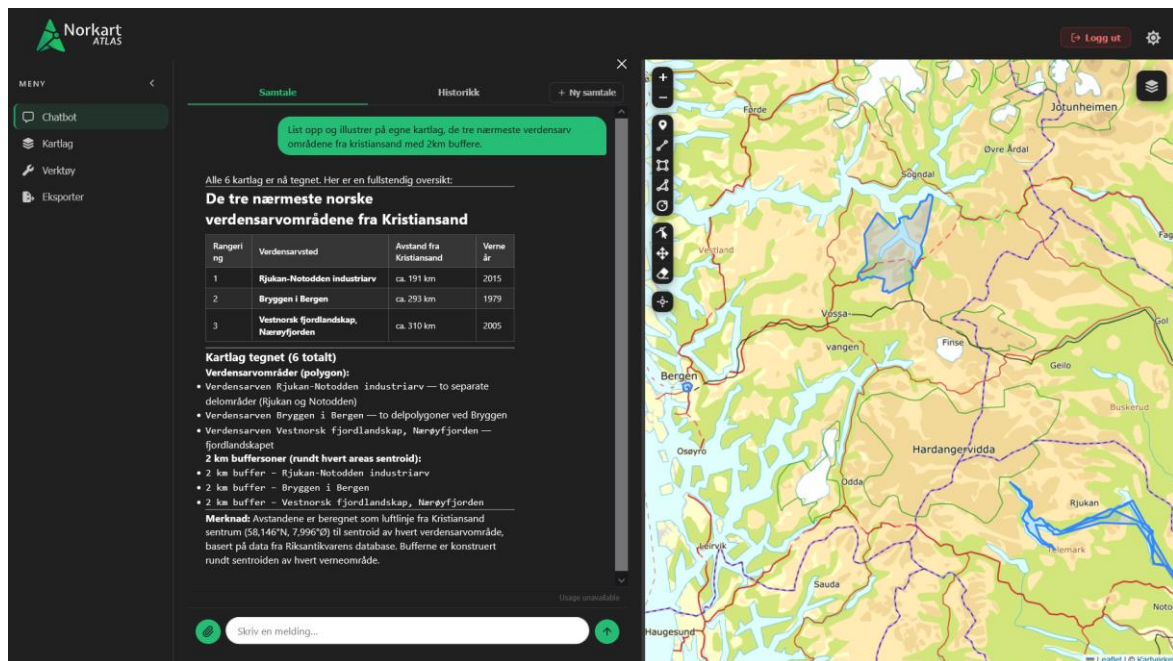
Teknologi og løsninger

Frontend er bygget i React med Leaflet som kartbibliotek. Kartlagene hentes som WMTS-fliser fra Kartverket og kan vises som topografisk, gråskalert, raster og nautisk kart. Brukere kan tegne og redigere i kartet via Leaflet-Geoman, og eksportere kartlag som GeoJSON, JSON, PNG eller PDF.

I backend kjører vi Python med FastAPI og Starlette, og strømmer svar fra KI-agenten til klienten via Server-Sent Events. KI-biten er to ting som spiller sammen: GitHub Copilot SDK håndterer selve sesjonen og strømmingen, mens FastMCP lar oss skrive egne verktøytjenere som KI-en kan bruke til å snakke med databasen, gjøre geografiske operasjoner, hente dokumenter og slå opp i vektorindeksen. Alt av data ligger i PostgreSQL med PostGIS, der vi bruker pgvector for semantisk søk og pg_trgm for fuzzy-matching. PDF-ene ligger i Azure Blob Storage, og vi bruker PyMuPDF til å trekke ut tekst fra dem når assistenten trenger noe konkret derfra.

Beskrivelse av produkt og resultat

Det gruppen sitter igjen med er en fungerende applikasjon hvor brukeren kan logge inn, åpne et kart, og føre en samtale med en AI-assistent som faktisk kan utføre handlinger. Når assistenten finner eller beregner noe relevant, dukker det opp som nye kartlag man kan slå av og på. Samtalene lagres slik at man kan komme tilbake til dem senere, og lagene kan eksporteres videre som GeoJSON, JSON, PNG eller PDF for bruk i rapporter, GIS-verktøy eller saksbehandling.



Figur 1 - Applikasjonen Atlas vises frem med en forespørsel og resultat

For å gi et inntrykk av hvordan det hele ser ut, har vi satt opp en GitHub Pages som viser applikasjonens tidligere stadier og funksjoner på <https://kartai.github.io/Atlas/>.

Hva vi er spesielt fornøyde med, og hva vi har lært

Overgangen fra Azure til GitHub Copilot SDK var beslutningen som ga mest utbytte for gruppen. Det var teknisk krevende å forkaste en fungerende prototype som allerede var utviklet og starte på ny arkitektur, men resultatet ble en mer stabil, modulær løsning med tilgang til ulike leverandører.

Samtidig var lærdommen rundt integrering av ulike KI-modeller opp mot egne MCP-servere svært nyttig, og utforskning av hvordan LLM-er er bygget opp ga gruppen mye ny innsikt som hjalp med oppsettet av et eget KI-basert system.

Den viktigste lærdommen for gruppen er at beslutninger bør tas tidligere og raskere enn instinktet tilsier. Både tilspissingen av prosjektomfanget og valget av riktig modell var situasjoner der vi i ettertid ser at vi ventet lenger enn nødvendig.

Til slutt, er gruppen fornøyde med at Norkart og Kartverket lot seg imponere av løsningen.