

Prosjektbeskrivelse

Knowit og Telenor Maritime

Beyond Dashboards: Agentic Observability for Maritime Incident Monitoring

Gruppe 9

Bachelorprosjekt i IT og informasjonssystemer ved Universitetet i Agder

Studenter:

Jonas Pettersen

<https://www.linkedin.com/in/jonas-pettersen-073931382/>

Kristian Bergedalen

<https://www.linkedin.com/in/kristian-magnus-bergedalen-221bb3186/>

Nidal Alendar

<https://www.linkedin.com/in/nidal-alendar-c9na0807/>

Pheeraphan Uthamontree

<https://www.linkedin.com/in/pheeraphan-uthamontree-4698a6382/>

Oppdragsgiver:

Knowit og Telenor Maritime

Prosjektet

Prosjektet undersøker hvordan maritime overvåkingssystemer kan gi bedre støtte når tekniske hendelser og driftsavvik oppstår. I maritime operasjoner finnes det store mengder data fra fartøy, applikasjoner og tekniske systemer. Tradisjonelle dashboards kan vise varsler, status og måleverdier, men de gir ikke alltid nok støtte til å forstå hvorfor noe skjer eller hva som bør undersøkes videre.

Målet med prosjektet var å utforske hvordan dashboards kan suppleres med observability-prinsipper, kontekst og KI-støttet analyse. Hensikten var ikke at prototypen skulle ta beslutninger, men at den skulle gi et bedre grunnlag for å undersøke hendelser og mulige årsaker.

Løsningen

Resultatet av prosjektet er en lokal kjørbare prototype for maritim incident monitoring. Løsningen er bygget rundt en UDS-applikasjonsovervåkingskontekst, der brukeren kan gå fra et varsel til mer detaljert informasjon om fartøy, applikasjon, logger, metric history og datakvalitet.

Kjernen er et hendelsesdashboard der brukeren kan følge et varsel fra oversiktsnivå ned til detaljer om påvirket fartøy, applikasjon, logghistorikk og datakvalitet. Løsningen ble bygget ut med støtte for flåteoversikt, som er en visning av hendelser på tvers av flere fartøy, og en undersøkelsesstøtte tilpasset driftssentre.

Prototypen inneholder også en AI-assisted analysis view (KI-støttet analysevisning). Når brukeren åpner en analyse for et varsel, henter systemet relevant kontekst og genererer en KI-støttet vurdering basert på tilgjengelige data. KI-en er ikke ment å ta beslutninger, men kan støtte brukeren med forslag til videre undersøkelser. En viktig del av løsningen er sporbarhet, der brukeren kan se hvilke verktøy og dokumenter som ble brukt som grunnlag for analysen.

Mot slutten av prosjektet utviklet gruppen også et dynamisk dashboard proof-of-concept (tidlig demonstrasjon av et hendelsestilpasset dashboard). Denne demonstrerer et mulig

oppsett for hvordan informasjon om en hendelse kan brukes til å lage en mer tilpasset dashboardvisning.

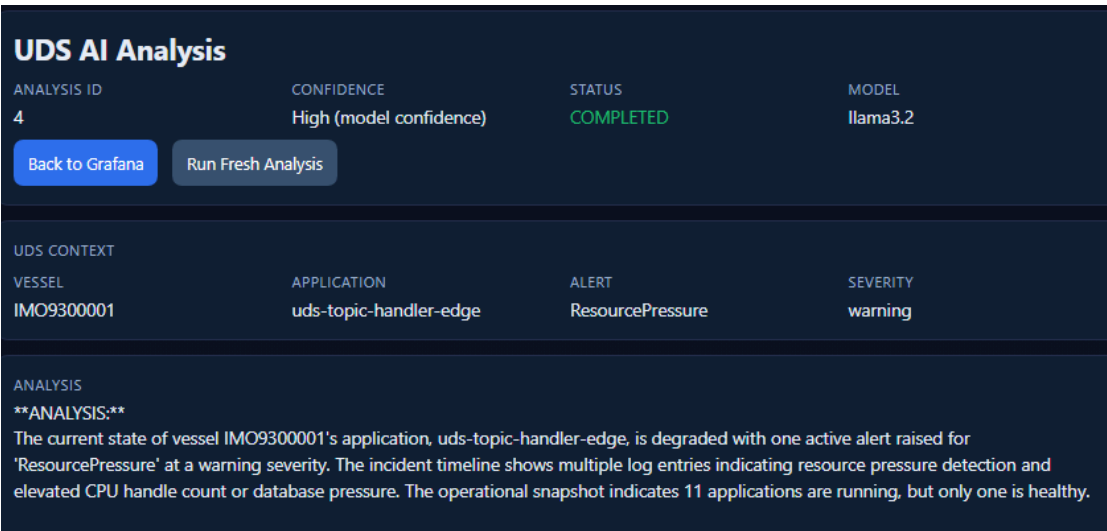


Teknologi

Prototypen er bygget med en lokal og åpen teknologistakk.

De viktigste teknologiene i prosjektet var:

- **Grafana** for dashboards og visualisering
- **PostgreSQL og TimescaleDB** for lagring av tidsseriedata, varsler, logger og metrics
- **Python og FastAPI** for backend og KI-integrasjon
- **Docker Compose** for lokal kjøring av systemet
- **Ollama** for lokale språkmodeller
- **pgvector og RAG** for dokumenthenting til KI-forklaringer
- **MCP-inspirert REST-adapter** for kontrollert tilgang til data og verktøy



Utfordringer

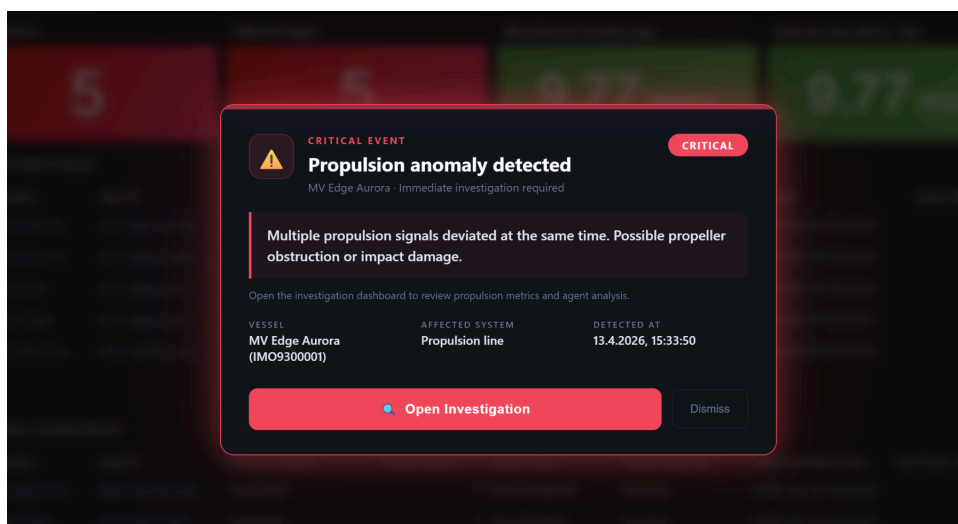
En av de største utfordringene var at prosjektet endret retning underveis. I starten arbeidet gruppen mer generelt med maritime dashboards, telemetri og sensorverdier. Etter hvert ble fokuset tydeligere rettet mot UDS-applikasjonsovervåking og incident-støtte. Dette krevde at gruppen justerte scope, prioriterte på nytt og sørget for at prototypen og rapporten hang sammen.

Prosjektet var også teknisk komplekst fordi det kombinerte flere områder som var nye for gruppen, blant annet observability, tidsseriedatabaser, Grafana, RAG, lokale språkmodeller og kontrollert KI-verkøytølgang. En viktig utfordring var å bruke KI på en ansvarlig måte, der løsningen er ment å støtte menneskelige vurderinger uten å erstatte dem.

Resultat og læring

Prosjektet ga viktig læring om scope control. Når man jobber med moderne teknologi som KI og observability, kan prosjektet raskt vokse. Derfor ble det viktig å skille mellom hva som var hovedleveransen, hva som var utvidelser, og hva som kun kunne demonstreres som proof-of-concept.

En viktig læring fra prosjektet er at IT-prosjekter ikke bare handler om å bygge funksjonalitet. De handler også om å forstå behov, håndtere usikkerhet, samarbeide med interessenter og ta tekniske valg som passer prosjektets faktiske rammer.



Lenker

GitHub:

<https://github.com/Code9-IT/Code9.git>

Demonstrasjonsvideo av prototypen:

<https://youtu.be/wPt3VBZrFWw>