

# IoT edge computing i Microsoft Azure

Jakob Simonsen  
Fredrik Frostad  
Michael Mobæk Thoresen  
Ole-Martin Heggen

24. januar 2020

# Innhold

<b>1</b>	<b>Presentasjon</b>	<b>3</b>
1.1	Oppgave . . . . .	3
1.2	Gruppemedlemmer . . . . .	4
1.2.1	Fredrik Frostad (s325853) . . . . .	4
1.2.2	Ole-Martin Heggen (s325905) . . . . .	4
1.2.3	Michael Mobæk Thoresen (s325903) . . . . .	4
1.2.4	Jakob Bogen Simonsen (s325908) . . . . .	4
1.3	Oppdragsgiver . . . . .	5
1.4	Kontaktpersoner . . . . .	6
1.4.1	Eva Hadler Vihovdle . . . . .	6
1.4.2	Kåre Langedrag . . . . .	6
<b>2</b>	<b>Sammendrag</b>	<b>7</b>
<b>3</b>	<b>Dagens situasjon (Status)</b>	<b>9</b>
<b>4</b>	<b>Mål</b>	<b>11</b>
<b>5</b>	<b>Rammebetingelser / Kravspec / Akseptansekriterier</b>	<b>12</b>
5.1	Rammebetingelser . . . . .	12
5.2	Kravspesifikasjon . . . . .	13
5.3	Akseptansekriterier . . . . .	13
<b>6</b>	<b>Løsninger</b>	<b>14</b>
6.1	Vår løsning . . . . .	14
6.2	Komponenter . . . . .	14
6.2.1	Del 1 - IoT . . . . .	14
6.2.2	Del 2 - sky . . . . .	15

6.2.3	Del 3 - presentasjon . . . . .	16
6.3	Arbeidsmetoder . . . . .	16
6.4	Arbeidsmetoder . . . . .	16
6.5	Prosess . . . . .	17
6.6	Tentativ tidsplan . . . . .	17
<b>7</b>	<b>Analyse av virkninger</b>	<b>18</b>

# Kapittel 1

## Presentasjon

### 1.1 Oppgave

Oppgaven går i hovedsak ut på å sette opp en IoT basert løsning i Microsoft Azure. Løsningen skal bestå av en komplett pipeline hele veien fra sensorer knyttet opp mot en edge node i form av en linux maskin og videre til dataene er lagret og prosessert i cloud. Et viktig design-kriterie for løsningen vil være å benytte så mange "off the shelf" Azure komponenter som mulig. Konfigurering og deployment av løsningen skal automatiseres slik at vi får en effektiv deployment-pipeline. Når løsningen er satt opp skal det gjøres kapasitetsmålinger med hensyn på hvor mye sensordata pr. tid som går gjennom løsningen. Det skal også gjøres en ende-til-ende cybersecurity vurdering av løsningen.

## **1.2 Gruppemedlemmer**

### **1.2.1 Fredrik Frostad (s325853)**

Fredrik(35) er dataingeniørstudent ved OsloMet og har tidligere arbeidet som prosjektleder og lydtekniker hos Bright Norway AS. Fra denne stillingen har Fredrik bred erfaring med prosjektering og gjennomføring av større prosjekter innen konserter, corporate-events og TV-produksjon. Fredrik har hatt sommerjobb som backend java utvikler hos Oslo Market Solutions, og jobber i dag deltid i samme bedrift. Fredrik starter som konsulent hos Solidsquare AS høsten 2020.

### **1.2.2 Ole-Martin Heggen (s325905)**

Ole-Martin(24) er dataingeniørstudent ved OsloMet og har tidligere utdanning som befal og dataelektroniker lærling i cyberforsvaret. Ole-Martin jobber i dag deltid hos Computas på avdeling for IT og analyse hvor han også har hatt sommerjobb. Stillingen hans innebærer systemadministrasjon, bruker supportering, Active Directory og Google Cloud Platform administrator. Han vil starte som konsulent for Sopra Steria høsten 2020. Ole-Martin har også interesse innenfor CTF og elektronikk.

### **1.2.3 Michael Mobæk Thoresen (s325903)**

Michael(21) jobber deltid som sikkerhetsanalytiker hos BDO og liker gaming. Michael har erfaring med Blue Team sikkerhet igjennom jobben hos BDO og litt erfaring med Red Team igjennom diverse CTF-konkurranser og aktiviteter som OverTheWire og HackTheBox.

### **1.2.4 Jakob Bogen Simonsen (s325908)**

Jakob(22) jobber som studentutvikler hos IBM og er studentassistent i faget programmering. Han har hovedsaklig jobbet med webapplikasjoner, distribuerte systemer og sky-løsninger. Siden hans 3. semester på dataingeniør studiet har han jobbet 70 prosent deltid og gjennomført prosjekter på alt fra maskinlæringsmodeller til verdikjede-løsninger med blockchain. Sist sommer hadde Jakob sommer internship hos IBM.

## 1.3 Oppdragsgiver

Kongsberg Digital er et selskap i Kongsberg Gruppen som leverer programvare og tekniske løsninger til gruppens kunder. Kongsberg digital ble etablert 17. mars 2016 og har i dag over 480 ansatte. Kongsberg Gruppen har i sin helhet over 11000 ansatte fordelt i gruppens verdensomspennende virksomheter. Kongsberg digital ble etablert for å fungere som et samlingspunkt for all digital kompetanse i Kongsberg Gruppen, og for å hjelpe de industrielle aktørene Kongsberg Gruppen leverer tjenester til med digital omstilling.

Kongsberg Digital leverer i all hovedsak teknologiske løsninger til kunder innen følgende områder:

- Handelsflåten
- Olje og gass
- Fornybar energi og kraftforsyning

Selskapets kjerneområder er IoT, kunstig intelligens, stor-data innsikt, maritim simulering, automatisering og autonome operasjoner. En av selskapets mest kjente løsninger er den digitale platformen kognifai. Kognifai gir kunder bla. tilgang til avansert datadrevet ressursstyring og Dynamisk Digital Tvilling teknologi. [Wikipedia, 2018][Kongsberg-Digital, 2020]

## **1.4 Kontaktpersoner**

### **1.4.1 Eva Hadler Vihovdle**

- Intern Veileder
- OsloMet
- epost: evav@oslomet.no
- tlf: 928 88 788

### **1.4.2 Kåre Langedrag**

- Prosjektansvarlig
- Kongsberg Digital
- epost: kare.langedrag@kdi.kongsberg.com
- tlf: 952 06 052

# Kapittel 2

## Sammendrag

I løpet av prosjektperioden skal vi sette opp, sikre, ytelsesteste og evaluere en IoT løsning basert på azure standardkomponenter. Vi vil basere oss på en referansearkitektur som benytter Azure PaaS (plattform-as-a-service) komponenter. Se figur 3.1 for eksempel på denne arkitekturen. Hovedformålet med prosjektet vil være og evaluere løsningens ytelse med hensyn på ende-til-ende dataflyt, sikkerhet og kostnad. Det er i all hovedsak tids-serialdata som produseres fra sensorene, og det er derfor viktig at vi velger en lagringsløsning som har god ytelse for denne typen data. I første omgang fokuserer vi på Microsoft CosmosDB, men vi er også åpne for å evaluere andre løsninger dersom dette lar seg gjøre innenfor prosjektets tidsrammer. Det mest aktuelle alternativet ser vi for oss at vil være PostgreSQL TimescaleDB.

Den innledende delen av prosjektet vil i stor grad bestå av research for å forstå platformen løsningen skal bygges i. Løsningen vil ha et høyt antall bevegelige deler", og det er svært viktig at vi oppnår god forståelse for virkemåte og interaksjon mellom alle de sammenknyttede komponentene.

I svært grove trekk kan prosjektet deles inn i følgende overordnede fokusområder:

- Automatisert deployment av løsning i Azure
- Evaluering av ytelse mtp. sensordata / tidsenhet
- Evaluering av sikkerhet i løsningen



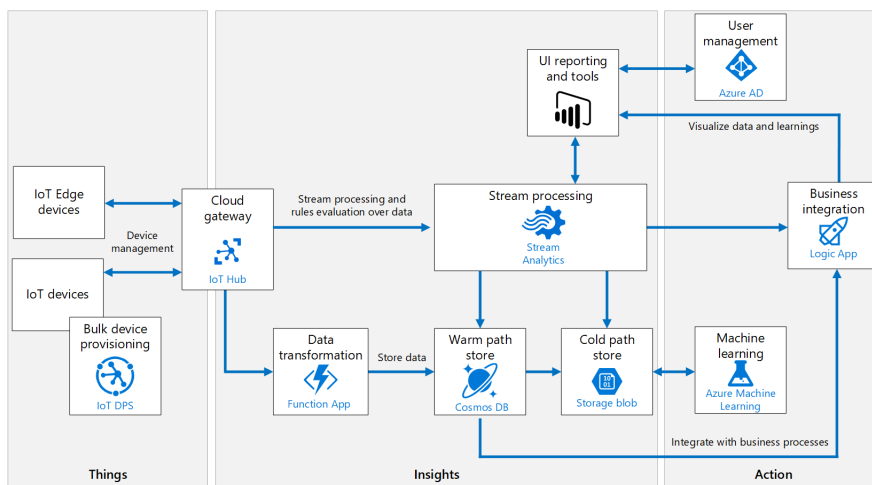
Alle gruppens medlemmer har erfaring med smidig utvikling fra tidligere prosjekter og nåværende arbeidsforhold. Vi vil derfor basere oss på en kombinasjon av SCRUM / KANBAN for å strukturere arbeidet gjennom prosjektet. Vi ser for oss en SCRUM sprint-basert arbeidsflyt i de første to tredjedelene av prosjektet, for deretter å gå over til KANBAN og taskbokning mot prosjektets slutfase.

# Kapittel 3

## Dagens situasjon (Status)

Kongsberg Digital AS (KDI) er i dag en verdensomspennende leverandør av industrielle IoT løsninger beregnet på å gi bedriftens kunder datadrevet innsikt. Løsningene som tilbys er basert på en kombinasjon av Azure skytjenester fra Microsoft og bedriftens egenutviklede løsninger. Microsoft sin løsning for mottak av IOT-data benyttes, mens løsningen for innsamling av data (IOT-Edge) og systemet for lagring og prosessering av data er egenutviklet av KDI.

I den konseptuelle tegningen nedenfor så vises de egenutviklede løsningene (som er relevant for denne) oppgaven med grønn bakgrunn.



Figur 3.1: Azure IoT reference architecture[Microsoft, 2020]

Når KDI ble stiftet i 2016 så hadde Microsoft akkurat lansert deler av sine IOT-løsninger, men viktige komponenter slik som selve IOT-Edge ble først lansert i 2018. Den første utgaven av IOT-Edge tilfredstilte ikke de kravene som KDI hadde til sin plattform, og det fantes ingen Cloud-tjenester som kunne lagre og prosessere tidsseriedata på en kostnadseffektiv måte.

Det har skjedd en stor utvikling på Microsoft sine IOT-løsninger de siste to årene, og for at KDI kan få et bedre beslutningsgrunnlag for å vurdere hvilke av Microsoft sine IOT-løsninger som potensielt kan erstatte egenutviklet programvare så ønsker de at det etableres en full ende-til-ende IOT-løsning basert kun på Microsoft Azure skytjenester.

En demonstrasjon av en slik ende-til-ende-løsning ansees som dekkende for å demonstrere om de funksjonelle kravene til løsningen er innfridd. Videre ønsker KDI at det gjøres målinger for å kalkulere ytelse pr kost for å kunne få et bedre grunnlag for å vurdere hvilke komponenter som eventuelt burde erstattes.

# Kapittel 4

## Mål

Det overordnede målet for prosjektet er å evaluere ytelse i en IoT edge computing løsning utelukkende basert på Azure standardkomponenter. Kongsberg Digital tilbyr i dag sin egen løsning for dette, men denne løsningen er en hybrid løsning med en blanding av selskapets interne stack og Microsoft Azure stacken. Dette er på grunn av at Azure sine standard komponenter på den tiden ikke levde opp til sikkerhets- og funksjonalitetskravene til Kongsberg Digital. Det er dermed interessant å se hvilke muligheter som ligger i en migrasjon til en løsning som baserer seg på en arkitektur bygget rundt konseptet Platform-as-a-service(PaaS). Det er viktig at deployment av løsningen automatiseres i så stor grad som mulig. Både gjennom skripting og kontainerisering av komponenter.

# Kapittel 5

## Rammebetingelser / Kravspec / Akseptansekriterier

### 5.1 Rammebetingelser

Implementasjon og evaluering av sikkerhet er en av de viktigste rammebetingelsene for prosjektet. Løsningen vil kunne eksponere potensielt kritisk infrastruktur og data mot internettet og det er derfor svært viktig å ha kontroll på at alle deler av løsningen er tilstrekkelig sikret. Kongsberg Digital ønsker å evaluere ytelesen til løsningen mtp. mengde sensordata behandlet pr. tidsenhet. Dette skal presenteres i et dashboard.

#### Tekniske rammer:

- Programmeringspråk: .NET C#, Bash, Javascript
- Versjonskontroll: Git
- Simulatorer for sensordata skal være containerbasert med bruk av LXD.
- Løsningen skal bygges med Azure komponenter
- Visualisering skal gjøres med Kibana PowerBI
- Prosjektstyring via Azure DevOps
- Teams og epost for kommunikasjon internt i Kongsberg Digital

- All tredjeparts programvare som benyttes må være lisensiert for kommersiell bruk
- Azure DevOps for dokumentasjon
- Deployment ved hjelp av Powershell, Azure Command Line Interface og Azure Resource Manager maler.
- WANem for simulering av nettverksforhold på autonome båter med begrenset internettilgang.

## **5.2 Kravspesifikasjon**

## **5.3 Akseptansekriterier**

# Kapittel 6

## Løsninger

### 6.1 Vår løsning

Kongsberg digital har i dag en egenutviklet IoT-plattform delvis basert på Azure komponenter. Vi skal lage en 100% Azure basert ekvivalent løsning for å kunne si noe om hvilken løsning som har de mest fordelaktige egenskapene. Løsningen deles inn i 3 forskjellige deler:

### 6.2 Komponenter

#### 6.2.1 Del 1 - IoT

Kongsberg tilbyr i dag en IoT løsning hvor de sanker inn sensordata og analyserer/lagrer det i skyen. Ved å lage en Azure basert løsning vil Kongsberg slippe en stor del vedlikehold av programvaren de tilbyr samtidig som det vil være mer generalisert teknologi som også kan brukes til andre formål.

I vår løsning tar edge-noden inn on-site sensordata for pre-prosесering og videresending til en hub som eksponerer tjenester i skyen. Videre sendes dataene til prosесering og lagring. Ved å bruke microsoft tjenestene er det enklere å videreutvikle løsningen med andre microsoft produkt integrasjoner. Kongsberg ønsker å øke sikkerhetsnivået og fleksibiliteten i deres løsning og vil derfor teste gjennomførbarheten av å gjøre en full microsoft implementasjon av deres nåværende løsning.

Vår edge node vil være en linux maskin som simulerer en naturlig dataflyt med for eksempel temperatur og trykk informasjon. Ved å simulere dataflyten vil vi enkelt kunne skalere antall enheter som vi tester systemet med. I et reelt system vil man raskt kunne få et stort antall enheter og for å kunne gjøre en fullverdig evaluering er det et viktig punkt å kunne se hvordan systemer håndterer dette. Dermed må systemet kunne skalere for å håndtere en svært høy data gjennomstrømning. Linux maskinen vår vil kommunisere med OPC UA (protokoll) over modbus (transport måte). Skalerings testene kan gjennomføres ved å spinne opp flere lettvekts virtuelle maskiner ved hjelp av LXD, eller ved multitrådede .NET kommandolinje applikasjoner.

IoT edge noden skal kunne gjøre prioriteringer på hvilken informasjon som må sendes/lagres. Siden ikke all informasjon er likestilt i et system kan det være nødvendig å ha mer kontroll på enkelte av sensorene og dette fører til at informasjon må prioriteres i køen. Enkelte sensorer som kontrollerer temperaturen vil ikke være så viktig med mindre det er snakk om unormalt høy temperatur eller trykk.

## 6.2.2 Del 2 - sky

Kongsberg er i dag en stor microsoft bruker og vi skal derfor basere oss på microsofts cloud computing platform Azure. Dagens løsning baserer seg på noen av de eldre delene av microsoft azure som ble tilbudt i det tidsrommet dagens platform ble utviklet.

Vår løsning skal da ta i bruk diverse standardiserte sky løsninger fra Azure som lagringstjenester (database løsninger), analytics (data prosessering og maskinlæring) og business logic. Disse tjenestene skal fungere i fullt samspill og gjøre alle de nødvendige databehandlingsstegene før lagring og visning.

Sky-laget skal også eksponere et API lag som kan brukes av ikke-tilknyttede tjenester som visningslaget eller videre prosessering som bruker for eksempel ikke vil ha lagret på skyen. APIene vil levere data i JSON format for lett videre prosessering og kontroll. API tjenesten åpnes som en nettside hvor man kan gjøre get forespørseler til hvis man har den medføl-



gende identifiserings tokenen som man får ved å gjøre en forespørsel mot autentiserings delen av systemet.

### **6.2.3 Del 3 - presentasjon**

Presentasjonslaget skal bestå av en nettside som viser dataflyt, dataforsinkelse og dataplot. Dette skal gjøres ved hjelp av kibana som er et datavisualiserings verktøy fra elasticsearch. Dette skal vises på en nettside som gjør requester til APIet i cloud løsningen og får data som skal visualiseres fortløpende.

## **6.3 Arbeidsmetoder**

Vi vil basere oss på smidig utviklingsmetodikk i prosjektperioden. Vi har alle god erfaring med dette fra tidligere, og mener at dette vil være riktig valg for et prosjekt i denne størrelsesordenen. Under de første 2/3 av prosjektet vil vi bruke SCRUM metodikk hvor vi definerer Epics som skal gjennomføres. Hver Epic vil igjen deles inn i en til to uker lange sprints der vi vil gjøre spring planning / retrospective i hver ende av sprinten. Basert på gruppens størrelse anser vi det som lite hensiktsmessig og holde fulle standups hver dag. Vi vil i stedet ha daglige korte statusoppdateringer, enten i plenum, eller via slack.

I prosjektets siste 1/3 vil vi gå over til å organisere arbeidet etter ren Kanban-metodikk. Dette gjør vi fordi det tempoet gjerne øker i sluttfasen i et prosjekt, og denne metoden underletter hurtig omprioritering av viktige oppgaver.

## **6.4 Arbeidsmetoder**

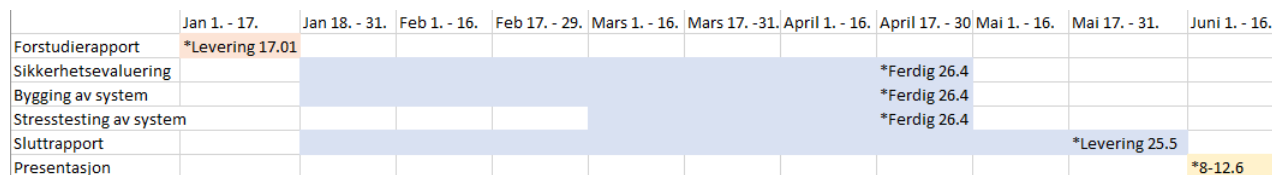
Vi vil basere oss på smidig utviklingsmetodikk i prosjektperioden. Vi har alle god erfaring med dette fra tidligere, og mener at dette vil være riktig valg for et prosjekt i denne størrelsesordenen. Under de første 2/3 av prosjektet vil vi bruke SCRUM metodikk hvor vi definerer Epics som skal gjennomføres. Hver Epic vil igjen deles inn i en til to uker lange sprints der vi vil gjøre spring planning / retrospective i hver ende av sprinten.

Basert på gruppens størrelse anser vi det som lite hensiktsmessig og holde fulle standups hver dag. Vi vil i stedet ha daglige korte statusoppdateringer, enten i plenum, eller via slack.

I prosjektets siste 1/3 vil vi gå over til å organisere arbeidet etter ren Kanban-metodikk. Dette gjør vi fordi det tempoet gjerne øker i slutfasen i et prosjekt, og denne metoden underletter hurtig omprioritering av viktige oppgaver.

Tidligere erfaringer med smidig utviklingsmetodikk har lært oss at det ikke finnes en "one-size-fits-all" løsning. Vi vil derfor løpende evaluere om det valgte rammeverket, det være seg SCRUM eller KANBAN er det rette valget for gruppen. Dersom vi erfarer at valgt metodikk virker mot sin hensikt vil vi reevaluere valget og velge en mere hensiktsmessig metode.

## 6.5 Tentativ tidsplan



Figur 6.1: Gant Diagram for Tidsplan

# Kapittel 7

## Analyse av virkninger

Den primære virkningen av dette prosjektet er å tillate Kongsberg Digital å evaluere nytten av å migrere deler av sine eksisterende IoT-tjenester over til azure standardkomponenter. Dersom resultatene av prosjektet skulle tilsi at en løsning bygget på denne platformen oppfyller de krav Kongsberg Digital har til kost, oppetid og sikkerhet, vil en slik migrasjon kunne frigjøre tid og ressurser brukt på vedlikehold av eksisterende platform. Dermed kan oppdragsgiver allokere en større andel tid og ressurser til utvikling og vedlikehold av sin forretningslogikk.

# Bibliografi

[Kongsberg-Digital, 2020] Kongsberg-Digital (2020). [På internett; besøkt 09-januar-2020].

[Microsoft, 2020] Microsoft (2020). Azure iot reference architecture.

[Wikipedia, 2018] Wikipedia (2018). Kongsberg digital — wikipedia,. [På internett; besøkt 09-januar-2020].