

# Forprosjekt

**Goodtech**

Christoffer-Theodoer Arntzen  
Victor Nascimento Bakke  
Tommy Abelsen

OsloMet - Storbyuniversitetet

<b>Presentasjon</b>	<b>2</b>
Goodtech Monitoring	2
<b>Sammendrag</b>	<b>2</b>
<b>Dagens situasjon</b>	<b>3</b>
<b>Mål og rammebetingelser</b>	<b>3</b>
Loggbehandlere	3
Databasen	3
Visualisering	4
Utviklingsprosessen	4
<b>Alternative løsninger</b>	<b>4</b>
Foreslått løsningsmodell	4
Potensielle alternativer	5
Loggbehandleren	5
Databasen	5
Visualiseringen	6
Utviklingsprosessen	6
<b>Analyse av virkninger</b>	<b>6</b>

# Presentasjon

## Gruppens medlemmer:

Tommy Abelsen s237393  
Christoffer-Theodor Arntzen s304920  
Victor Nascimento Bakke s237590

## Veileder:

Boning Feng  
OsloMet - Storbyuniversitetet  
[boningf@oslomet.no](mailto:boningf@oslomet.no)  
Mobil: 95897687  
Tlf: 67238714

## Oppdragsgiver:

Goodtech AS  
Innspurten 15, 0663 Oslo

## Kontaktperson:

Mikkel Sannes Nylend  
Goodtech  
[mikkel.sannes.nylend@goodtech.no](mailto:mikkel.sannes.nylend@goodtech.no)  
Mobil: 99426162

## Goodtech Monitoring

Prosjektet handler om normalisering av loggdata fra Goodtechs Manufacturing Execution System (MES)-systemer som sitter ute på kundens lokaler. Denne dataen skal bli lastet opp i skyen og deretter visualiseres på en web løsning. Den største utfordringen blir å lage en løsning som normaliserer dataen fra MES-ene og sender dette til visualiseringsplattformen. Det er ønsket av Goodtech at gruppen selv bestemmer hvilke data som skal bli logget, men det var ønsket at StatusLogEntry-tabellen og QueueElement-tabellen blir prioritert i startfasen. Ønsket til Goodtech er at dette prosjektet skal bli en tilleggsmodul i deres MES-systemer.

## Sammendrag

I denne rapporten ser vi på oppgaven vi har fått fra Goodtech. De ønsker at vi lager en monitoreringsløsning for deres MES-systemer. Goodtech har per i dag ikke noen måte å oppdage feil eller mangler på systemene deres ute hos kundene. Kravene Goodtech stiller til løsningen er at studentene skal lage en loggbehandler som sitter på systemene ute hos kundene, som sender loggdata til en sky-basert database. Disse dataene blir så lest av en web-basert visualiseringsløsning som presenterer dataene på en visuell og intuitiv måte. Fremgangsmåte i utviklingen blir smidig der gruppen skal planlegge hva som skal gjøres ukentlig. Vi har så en liste over potensielle alternative løsninger og til slutt en analyse av virkningen denne løsningen vil ha på Goodtech sin virksomhet.

# Dagens situasjon

Per i dag har Goodtech ingen oversikt over loggene fra Manufacturing execution system (MES) hos kundene i sitt bedriftslokale og belager seg på at kunden tar kontakt ved en feil.

Dagens situasjon kan dermed føre til at feil i systemet ikke oppdages. Dersom en feil ikke oppdages over en lang periode kan dette føre til store økonomiske konsekvenser, grunnet tilbakekalling av produkter, skade på systemer, mangelfulle bestillinger eller lignende.

Når en kunde først tar kontakt med Goodtech om en potensiell feil, hender det at kunden unnlater å nevne egen feilsøking. Dette kan føre til at Goodtech bruker ressurser på feilsøking som allerede er gjort, eller at kunden har kommet i skade for å lage nye feil under egen feilsøking.

Hensikten med Goodtech Monitoring er å få tilgang til visuell data. Visualiseringen av loggdata gjør det enklere å følge med på når en feil oppstår, eller forvarsler på hva som kan føre til feil i systemene.

Med Goodtech Monitoring kan Goodtech enkelt og raskt sette i gang feilsøking når det oppdages feil eller potensielle feil, selv uten at kunden opplever dette eller varsler selv. Goodtech Monitoring vil også enklere identifisere hvor feilen ligger, slik at man ikke må grave seg gjennom unødvendig mye loggdata for å finne kilden til problemet.

## Mål og rammebetingelser

Oppgaven skal i all hovedsak bestå av fire komponenter:

- Loggbehandlere ute på kundenes servere som sender loggdata til skyen
- Database i skyen
- Visualiseringsplattform som leser og tolker data fra databasen
- Utviklingsprosessen

## Loggbehandlere

- Leser data fra databasen til MES-systemet til kunden.
- Vi utvikler våre løsninger mot siste versjon av systemet til Goodtech, så hver kunde har generelt sett samme struktur på loggdataene.
- Leser data fra StatusLogEntry-tabellen.
- Leser data fra QueueElement-tabellen.
- Skal kun være tilgjengelig på Goodtechs nettverk og for hver individuelle kunde.

## Databasen

- Vi har behov for en kjapp database som håndterer store mengder data, så vi går for Microsoft Azure Database (MSSQL) i Microsoft Azure Cloud, en Database as a Service (DBaaS)-løsning for databaser i skyen.

- Skal kun leses fra Goodtechs nettverk, men skal kunne skrives til fra kundenes nettverk.

## Visualisering

- Web-basert visualisering som presenterer dataene, og muliggjør konklusjoner basert på dem. Slik som hvorvidt en kunde sin system-tilstand er god, om det har forekommet en feil hos en kunde som må tas hånd om manuelt osv.
- Grafana på en egen Azure-instans.
- Benytter regler for tolking av data fra StatusLogEntry-tabellen, som inneholder løpende hendelser i systemet.
- Benytter regler for tolking av data fra QueueElement-tabellen, som er selve kø-tabellen systemet bruker for hver ting som skal gjøres.
- Skal kun være tilgjengelig på Goodtechs nettverk.
- Benytter Microsoft Azure Cloud.

## Språk

- Java
- HTML
- CSS
- JavaScript
- SQL

## Verktøy

- IntelliJ
- Grafana
- MSSQL studio
- Subversion
- Google Drive

## Utviklingsprosessen

Gruppen vår har planlagt å benytte en smidig arbeidsmetode. Dette er fordi vi har lite forhåndskunnskaper om dette prosjektet, og mye av informasjonen vi trenger vil komme underveis i arbeidet. Samtidig ønsker kunden at vi, på vår egen måte, finner ut av hvilke data som kan være nødvendig å inkludere i visualiseringsløsningen. Med en smidig arbeidsprosess vil det lettere tilrettelegges for at vi gjør endringer underveis i arbeidet, som å inkludere mer nødvendig data, eller å ekskludere tjenester eller unødvendige logger. Samtidig vil oppdateringer og funksjonalitet komme fortløpende, noe som gjør at kunden kan være med på å påvirke prosjektet underveis.

# Alternative løsninger

## Foreslått løsningsmodell

Lage en loggbehandler som blir plassert på Goodtechs server ute hos kunden. Den vil der finne og behandle relevante logger. Deretter vil loggene bli sendt opp til skyløsningen vi skal sette opp. I skyløsningen vil loggene videre behandles og settes inn i en database. Databasen lagrer logger i en viss periode. Vi bruker Grafana for å visualisere dataene loggene inneholder. Visualiseringen i Grafana vil gjøre loggdata synlig, og vil varsle om ønskede hendelser visuelt og rimelig raskt.

Valg	Fordeler	Ulemper
Grafana	Trenger ikke lage eget dashboard.	Ikke like skreddersydd eller tilbøyelig.
Azure SQL Database	Trenger ikke sette opp egen server på Goodtech sitt eget nettverk.	Har ikke like mye kontroll over serveren til databasen.
Parsing i Grafana	All tolking og tyding av dataene skjer på ett enkelt sted. Visualiseringen har kontroll på hvilke data som er viktige eller ikke. Enklere loggbehandlere som bare henter og sender data.	Kan danne køer og flaskehals. Unødvendig data som blir lagret i databasen.
Java	Vi har fra tidligere i studiene brukt Java. Goodtech bruker Java.	I denne sammenhengen er det i praksis ingen direkte ulempe med å bruke Java, gitt at det er det språket vi kan til felles.

## Potensielle alternativer

### Loggbehandleren

Kravene til Goodtech er så spesifikke for loggbehandleren at det er i grunn ingenting vi kan gjøre annerledes med dem. Alternativt så kunne de forhåndsnormaliserte data før de ble sendt til databasen, men det ender opp med at vi må tolke dataene på visualiseringssiden uansett, slik at dette ikke er noen gevinst for oss.

### Databasen

Alternative databaser:

- Elasticsearch: Vi tenkte originalt på å bruke Elasticsearch, men siden Elasticsearch må settes opp manuelt i Azure valgte vi heller en enklere løsning.
- Cosmos DB: Vi valgte først Cosmos DB som en integrert Azure database-løsning, men innser i ettertid at Grafana har ikke integrasjon mot Cosmos DB. Dette kan løses ved å ikke bruke Grafana, eller skrive et plugin til Grafana for å integrere mot Cosmos DB, men det viste seg å være lettere å heller velge Azure SQL Database, som er en database hostet av Azure.

Vi kunne også hatt databasen i Amazon Web Services istedenfor Microsoft Azure, eller eventuelt hatt den på en dedikert server på Goodtech sitt nettverk.

### Visualiseringen

Vi har valgt å gå for Grafana fordi det ble anbefalt av Goodtech, og etter litt research så har vi funnet ut at dette er en plattform som passer greit for dette prosjektet. Alternativer er å lage dashbord selv, og benytte verktøy som D3.js for å generere grafer fra dataene vi henter inn fra databasen. Kibana er også et alternativ som blir mye brukt i sammen med Elasticsearch.

### Utviklingsprosessen

Vi har planlagt å skrive oppgaven i Java, med HTML, CSS og Javascript frontend i visualiseringen. Vi har erfaring med C# som alternativer til Java, og eventuelt Typescript som alternativer til Javascript, ennå det er et språk som transpileres til Javascript uansett.

## Analyse av virkninger

Prosjektet skal gjøre Goodtech mer bevisst på deres kunders systemer. Det er ment at dette prosjektet skal gjøre Goodtech bedre på å respondere til feil i systemer hos kundene og kutte ned på tiden fra feil blir oppdaget til at feil blir løst.

Tidligere har det blitt nevnt at det har tatt en måned fra feil har blitt oppdaget av kunden til at det ble varslet til Goodtech. Med denne monitoreringsløsningen vil ikke Goodtech lenger være avhengig av at kunder selv varsler om feil.