

Prosjektdagbok - Biogrid

Peter Wilhelmsen

February 19, 2016

Innholdsfortegnelse

1	Introduksjon	3
2	Dagbok	3
2.1	November 2015	3
2.2	Desember 2015	3
2.2.1	Møte med Biogrid 14.12.15	3
2.3	Januar 2016	4
2.3.1	Forprosjektrapport 07.01.16	4
2.3.2	12.01.16	4
2.3.3	14.01.16	4
2.3.4	Videomøte med Biogrid 15.01.16	4
2.3.5	Møte med veileder Thor Hasle 21.01.16	5
2.3.6	22.01.16	5
2.3.7	Kravspesifikasjon 25.01.16 - 03.02.16	6
2.4	Februar	6
2.4.1	Møte med Biogrid på Startup Lab - 04.02.16	6
2.4.2	JavaScript bibliotek for grafer 05.02.16	8
2.4.3	Evaluering og valg 09.02.16	8
2.4.4	Meteor i kombinasjon med ReactJS15.02.16	9
2.4.5	Fra database til graf 18.02.16	10
2.4.6	Latex og graf 19.02.16	10
2.4.7	Fra testFil	10

1 Introduksjon

Våren 2016 gjennomfører gruppe 20 bachelorprosjekt på Høgskolen i Oslo og Akershus i samarbeid med Biogrid Cortex AS.

Dagboken er delt opp i måneder, som igjen er delt opp i underpunkter.

2 Dagbok

2.1 November 2015

Tok kontakt med Biogrid på mail og Oscar er i møte med August Flatby som er daglig leder for Biogrid AS. En uke senere er vi igjen i kontakt med Biogrid og de ønsker å samarbeide med oss. Gir oss tid til å tenke på tilbudet, men vi har ingen med å takke ja til tilbudet. Avtaler å ha møte 14. desember i Biogrids lokaler når Peter er tilbake i Oslo.

2.2 Desember 2015

2.2.1 Møte med Biogrid 14.12.15

Oscar og Peter i møte med August Flatby. Får presentert prosjektet og ideer Flatby har. Får innføring om diverse teknologier som Biogrid kan tenke seg skal brukes. Disse er:

- Rickshaw.js for grafisk fremstilling
- Bootstrap
- Meteor
- MongoDB

Biogrid gir beskjed om at vi står fritt til å komme med forslag selv til hva som kan brukes. Avtaler å møtes igjen over nyttår.

Tiden frem mot nyttår går med til å se litt på aktuelle pakker i JavaScript platformen Meteor.

2.3 Januar 2016

2.3.1 Forprosjektrapport 07.01.16

Møtes hos Haakon der vi diskuterer litt og går gjennom hva som skal være med i en forprosjektrapport.

Sender mail til Biogrid der vi ber om møte. Får svar dagen etter. Han er opptatt med å lage til oppsett på hvordan dataen skal lagres, ingen respons på forespørsel om møte.

2.3.2 12.01.16

Gruppen møtes på skolen. Fortsatt ikke fått avtalt møte med Biogrid.

Gruppen diskuterer bruken av GitHub og hvilken editor vi skal bruke til rapport-skriving. Alternativene er Word Online, Pages og Google Docs. Gruppen blir enig om at Word er det beste alternativ da dette er editoren som har mest funksjonalitet tilgjengelig.

2.3.3 14.01.16

Gruppen møtes hos Haakon. Sender ny mail til Biogrid for å avtale møte.

Oppretter dokument for forprosjektrapporten, som foreløpig kun har vært notater. Gruppen kommer et godt stykke med rapporten, venter kun på Biogrid for å avtale endelige betingelser, mål og lignende.

2.3.4 Videomøte med Biogrid 15.01.16

Har videomøte med Biogrid på <http://www.appear.in>. Tar opp noen punkter vedrørende forprosjektrapporten. Hele gruppen deltar. Har på forhånd sendt utkastet av rapporten til Biogrid. Det er gjort noen små endringer, men ellers fornøyd med hvordan den er strukturert.

Oppsummering av hovedpunkter:

Løsning består av:

- IBM Softlayer
- Docker: Løsningen deployes i Docker

- Node-RED: data stream processor
- Mosquitto, MQTT Broker
- MongoDB
- Meteor: selve applikasjonen
 Pakkebruk i Meteor. Mange fine løsninger, men redd det kan bli en for stor snarvei. Samtidig er det mye kode å sette seg inn i og finne ut hvordan fungerer. Tar dette opp veileder.

Vet ikke hvor nøye det er hvor mye vi forklarer hvert enkelt hovedpunkt i løsningen siden vi konsentrerer oss om Meteor og MongoDB. Skissen vi fikk tilsendt passer fint inn her, men da kanskje greit å lage en kort forklaring til den.

Det skal skrives kontrakt mellom skolen og oppdragsgiver. Gruppen skal kontakte veileder for møte. Foreløpig ingen informasjon hvordan kontrakten fungerer. Gir beskjed til Biogrid etter møte med veileder.

2.3.5 Møte med veileder Thor Hasle 21.01.16

Oscar og Peter i møte med gruppen veileder, Thor Hasle. Presenterer prosjektet og får innspill fra Thor.

- Ingen faste møter med Hasle. Ta kontakt når vi trenger, kan bruke han som konsulent.
- Kontrakt mellom arbeidsgiver og HiOA: bruke standardkontrakt som finnes på bachelorsiden.
- Begynne på kravspec og arbeidsplan vil være neste steg.
- Tips: definere Internet of Things i rapporten.

2.3.6 22.01.16

Gruppen og Biogrid bruker Slack som kommunikasjonsmiddel. Dette er en applikasjon som gjør det lett å sende meldinger, dele filer og lignende.

August sender over linker om ReactJS. Det er JS bibliotek som brukes til å lage UI-komponenter. Det er spådd at React er up and coming, og vi får to artikler å lese. React virker bra å bruke og gruppen starter å se på tutorials og sette seg inn i biblioteket.

Sender over kontrakten mellom skolen og Biogrid, og vi avtaler møte 4.februar når de er på plass i nye lokaler på Startup Lab i Forskningsparken.

2.3.7 Kravspesifikasjon 25.01.16 - 03.02.16

Møtes på skolen for å starte på kravspesifikasjonen. Diskuterer hvilke funksjonelle og ikke funksjonelle krav, milepælsplan. Vanskelig å sette noe veldig konkret da det fortsatt trengs innspill fra Biogrid om ønsket funksjonalitet osv.

Gruppen må få innføring i hvordan Docker og IBM Softlayer fungerer. Gruppen ønsker også en kjapp innføring i bruken av Github da den ikke føler seg helt komfortabel med bruken av versjonshåndteringssystemet.

2.4 Februar

2.4.1 Møte med Biogrid på Startup Lab - 04.02.16

Hovedpunkter i møte:

- Softwaren har gått fra å være tiltenkt kun innen landbruk til å kunne brukes i flere hold. Skal utvikles generisk slik at det kan passe flere formål. Det finnes lignende løsninger, men vi skal ikke skremmes av at vår løsning vil ligne/gjøre det samme som dem. Denne løsningen skal ikke stagnere, men kunne utvikles videre.
- Hvem skal bruke systemet?
Målgruppen er fortsatt ikke helt sikker. Det er IKKE ment for hobbybruk, men for en industri som for tiden ligger an til å være innen landbruk, jordbruk eller lignende. 1. mars skal Biogrid ha seminar for bedrifter innen landbruk.
- Arkitekturen
Docker er som en virtuell maskin, men kalles en container. Det er én container pr prosess der hver container tror den er en maskin. Containerne har hvert sitt filsystem og hersker over seg selv. Containerne kan også dele filsystem. En container kan tas ned uten å påvirke de andre. Containerne kan snakke med hverandre. Docker Compose brukes til å starte containerne samtidig.(script). Docker lokalt er helt likt som på en server, veldig lett å sette ut i virkeligheten uten modifikasjoner.

BiogridCortex består av: - Cortex: orkestrer alt sammen.
Mongo

Mosquitto

NODE-RED

Meteor. (foreløpig et lite spørsmåltegn til bruken av plattformen).

MQTT BROKER Hvis noen er interessert i disse dataene sendes de videre.

Har en slags adressering. Du sier ikke hvem du er, men hvor du vil sende dataene. TOPIC: /sensor/{type} /sensor/co2 payload: ID:GUID (Global Unique ID) VALUE: 413.6

- Milpæler

1. Visualisere data
2. Liste av sensorer
3. Logge inn - få mulighet til å velge hvilken lokasjon du vil se data for

- Fremdrift videre

Sette opp arbeidsmiljø

Velge editor. Biogrid mener Atom er et godt alternativ.

GitHub. Bruk Sourcetree, mange mener GitHub Desktop er for simpelt.

Sjekke forskjellige bibliotek for grafer. Sammenlign og finn beste alternativ.

UX. Biogrid mener Bootstrap er et godt alternativ. Gruppen har brukt det litt før og er enig.

Implementasjon

Deretter begynne å gruppere sensorer og bygg/lokasjoner

User account. Vil ikke bruke tid på å logge inn og ut under utvikling

Oppsummering:

Fikk satt opp arbeidsmiljø(Docker) og August viste oss hvordan dette funket. Hele gruppen er online og kan starte generering av data til MongoDB. Fremover vil vi lese oss opp på forskjellige alternativer med tanke på fremstilling av data: fokus på interaktivitet, estetikk, må kunne oppdatere data i realtime og skal ikke være kommersielt.

Det må leses opp om Docker og MongoDB.

Største vending/overraskelse: Biogrid har begynt å tvile på Meteor som plattform. Meteor bygger sammen alt for deg, som gir mindre kontroll. Vil at gruppen sjekker opp andre webteknologier for å se om det finnes andre gode alternativer. Tenk også på hva gruppen har lyst og er interessert i å lære. Meteor er IKKE utelukket.

2.4.2 JavaScript bibliotek for grafer 05.02.16

Gruppen møtes for å finne ut av hvilke biblioteker for fremvisning av grafer som er aktuelle. Gruppen har lest artikler om anbefalte biblioteker:

<http://thenextweb.com/dd/2015/06/12/20-best-javascript-chart-libraries/>
FLERE LINKER*

Å kunne zoome er en viktig funksjon gruppen ser etter. Det må være mulig å kunne zoome seg inn fra dag, til time og til minutt. De tre aktuelle bibliotekene er:

- Dygraphs: et open source JavaScript bibliotek som lager grafer. Det er interaktivt slik at det er mulig å forandre grafen ved å zoome inn på ønsket periode. På papiret ser det veldig aktuelt ut. Rent designmessig ser det ikke så bra ut, men vi regner med at dette er noe som kan forandres på senere.
<http://www.dygraphs.com>
- Flot: et JavaScript bibliotek for jQuery. Også interaktiv som gjør det mulig å zoome inn på grafen for å få et mer detaljert bilde. Brukt av veldig mange forskjellige bedrifter. God dokumentasjon, med video blant annet. Ser kanskje litt bedre ut en dygraphs. Siden det bygger på jQuery er de fleste browsere også støttet. Støtter JSON. Virker enkelt å lære, men kanskje det blir for enkelt for oss. <http://www.flotcharts.org>
- Rickshaw: Javascript rammeverk som er open source. Ble anbefalt dette av arbeidsgiver første gang, men med forbehold om at det kan finnes bedre alternativer der ute. Muligheter for interaktive grafer her også. Støtte for charts med time-series data. Reklamerer med at de kan lage grafer basert real time oppdatering. Bygger på D3 som skal være et godt rammeverk.
<http://code.shutterstock.com/rickshaw>

2.4.3 Evaluering og valg 09.02.16

Frem mot i dag har tiden gått med på å lese om React, finne ut om Meteor er godt nok til å være gruppens webplattform og sjekke de grafiske JS bibliotekene og lese om Docker.

- Etter møte forrige uke var August litt betenkt på bruken av Meteor, men i dag virker det som han tror det vil funke bra. Han sender oss en link på bok/pdf om Meteor som han mener kan være en god ressurs. Dette tar vi som et tegn på at han syns gruppen skal gå for Meteor! Peter har funnet artikkel (<http://tutorialzine.com/2015/12/the-languages-and-frameworks-you-should-learn>) om at Meteor vil være en platform som er "up and coming" i 2016. Leter

fortsatt etter flere artikler som understøtter denne påstanden. Sier at Meteor er en hybrid mobile framework og at det er verdt å lære.

- Valg om editor lander på Atom for alle tre. Atom har flere tilleggspakker man kan installere for å optimalisere og tilpasse bruk:
 - Language støtte for ReactJS: `jsx`.
 - Bootstrap autocomplete. Kan være denne ikke vil fungere sammen med ReactJS pakken pga class ikke brukes i html-tagger i react, det er erstattet med `className`.
- August har gjort endringer på hvordan data legges inn i collection i MongoDB. I stedet for at det skal være en collection for hver sensor type, er det nå 1 collection som heter sensor, der hvert dokument har sensor-id, day og type: rh, co2. Endringene er pushet til GitHub og funker for gruppen.

2.4.4 Meteor i kombinasjon med ReactJS15.02.16

- Biogrid har sett på alternative og foreslår å bruke Dygraphs. Dette var den gruppen likte best. Har flere funksjoner, men ser ikke veldig pent ut. Har sett på docs om CSS styling, og det skal være mulig å tilpasse slik gruppen måtte ønske.
- Biogrid har sett mer på webstacken og selve React. Sier React virker veldig bra. Sender link på video som forklarer hvorfor Facebook utviklet react: <https://facebook.github.io/flux/docs/overview.html> (se fra 24.00 ca).
Viktig: React brukes kun for å lage UI-komponenter.
- Biogrid mener Meteor er en god måte å holde data syncet mellom server og klient, så det vil satses på en Meteor-React kombinasjon.
- Gruppen har prøvd seg på tutorialer om React: <https://www.youtube.com/watch?v=kVbVBp35keQ>. Denne tutorial bruker Flowrouter og react-layout, to pakker fra Kadora som er "performance monitoring for meteor" - <http://www.kadora.io>. Dette virker som to gode pakker for å håndtere routing og hvordan man bygger opp layouter i React. Gruppen sjekker fortsatt alternativer til routing og hvordan man kan bruke react. Slik det ser ut finnes det flere fremgangsmåter.

2.4.5 Fra database til graf 18.02.16

- Haakon og Oscar har startet med å hente data fra MongoDB og sette disse inn i graf ved hjelp av Dygraph. Pushet TestKode til GitHub. Peter får kjørt Meteorprosjektet på sin maskin. Planen er at Oscar og Haakon fortsetter på dette i morgen.
- Peter er hjemme i Ålesund. (11.02 - 25.02(?)). Har lest gjennom AirBnb JS styleguide, <https://github.com/airbnb/javascript>, og leser om React. Skal lage oversikt over pakker og alternativer på hvordan React skal brukes.
- Haakon funnet Robomongo. Applikasjon som man kan koble opp mot MongoDB. Oversiktlig GUI, der man lett kan se de forskjellige documents og gjøre spørringer. Slepper å gjøre spørringer i terminal.
- Haakon har gjort oppdatert hjemmesiden til prosjektet. Lastet opp forprosjektrapport.

2.4.6 Latex og graf 19.02.16

Peter har testet ut Latex som teksteditor for rapportskriving. Latex gjør det lett for gruppen å jobbe på samme dokument, og man trenger ikke å logge seg på Word Online. Latex gir ryddig dokumentstruktur og det er lett å generere overskrifter, bilder/figurer og innholdsfortegnelse.

Haakon og Oscar jobber videre med spørring mot databasen.

Problemstillinger: Hvilken form for routing skal vi bruke? Peter har lest opp om Flow-router og React-router, begge ser ut til å være gode alternativer. Det finnes mer dokumentasjon på React-router, men Flowrouter har blitt brukt av mange i kombinasjon med Meteor. Lest denne linken: <https://forums.meteor.com/t/react-router-vs-flow-router/6867>

Link til ReactJS Github med React-router <https://github.com/reactjs/react-router>.

2.4.7 Fra testFil

Sjekk hvordan denne legger seg i TOC. Innholdet fra denne filen kommer fra test-Fil.tex

- test
- test