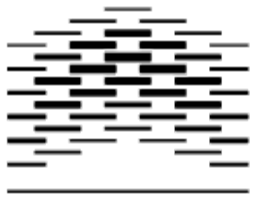
goodtech

Forprosjektrapport gruppe 23

Shahin Jahangiri, Tobias Johansen, Bjørnar Birkeland, Martin Sivertsen



HØGSKOLEN I OSLO
OG AKERSHUS

Presentasjon

Tittel på prosjekt	Goodtech API for IoT
Gruppemedlemmer:	Bjørnar Birkelands s236309 Shahin Jahangiri s198732 Martin Sivertsen s236317 Tobias Johansen s236335.
Oppdragsgiver:	Goodtech Projects & Services AS, Per Krohgs vei 4A 1065 Oslo http://www.goodtech.no/
Kontaktperson:	Harald Pedersen, Avd.leder Industriell IT harald.pedersen@goodtech.no 99 09 51 79
Intern veileder:	Aiko Yamashita, Førsteamanuensis aiko.yamashita@hioa.no 47 45 12 42
Midlertidig prosjektside:	http://student.cs.hioa.no/~s198732/Goodtech/

Sammendrag

Gruppe 23 ved Høgskolen i Oslo og Akershus skal utarbeide et system for det nordiske teknologikonsernet Goodtech. Goodtech ønsker et API[1] mot Eclipse Milo for å hente og sende data, søke etter objekter/maskin-adresser og abonnere på endringer i data. De ønsker også en web-klient som bruker dette API'et for å kommunisere med produksjonssystemene. Firmaet har allerede en fungerende løsning, men ønsker et forbedret grensesnitt og implementasjon av en ny stack[2]. Teknologiene som skal brukes er satt på forhånd av Goodtech, men gruppen kan selv velge hvordan de blir brukt.

Dagens situasjon

Goodtech er et nordisk teknologikonsern som leverer prosjekter, tjenester og produkter innen elektro- og prosessteknikk, miljøteknikk og industriteknikk til energi, industri, infrastruktur, bygg og offentlig sektor i Norden. Goodtech tilbyr en tjeneste som lar kundene bruke en web-klient for å kommunisere med industrielt utstyr og kontrollere/samle opp data. Dette systemet bruker en gammel versjon av en stack som tar i bruk protokollen OPC-UA[3]. Goodtech ønsker å bytte ut denne med en ny OPC-UA stack kalt Eclipse Milo, som er open-source og skrevet i ren Java. Ved å bytte til Eclipse Milo vil deres tekniske gjeld[4] minke, da det blir lettere å legge til ny funksjonalitet i fremtiden. For at det skal være mulig å bytte ut stacken, må en ny web-klient utvikles og et API som kan hjelpe den å kommunisere med Eclipse Milo. Oppgaven går derfor ut på å utvikle disse slik at Goodtech kan bytte over til den nye stacken. Gruppen har mye erfaring med programmeringsspråket som skal benyttes, men må fortsatt bruke teknologi ingen har kjennskap til og må derfor sette seg grundig inn i disse før utviklingsfasen kan starte.

Mål og rammebetingelser

Målet med systemet er å samle data fra logiske kontrollenheter, og gjøre dette tilgjengelig for brukeren på en lett og oversiktlig måte på Windows og Linux maskiner. Brukerne skal kunne logge inn på en webapplikasjon som gir oversikt over dataen. Et eksempel på bruksområde er et ventilasjonssystem i en bygning, som blir styrt av en kontrollenhet som er koblet til en server. En kan da for eksempel observere diverse temperaturer, endre termostater eller endre lufttilførsel. For at Goodtech skal ha bruk for systemet som utvikles, er det viktig at produktets funksjonalitet ikke er dårligere enn funksjonaliteten til den nåværende løsningen. Forutenom minimumskravet er det ønsket tilleggsfunksjonalitet som søk på

objekter/maskin-adresser og abonnering på data. Systemet skal kommunisere med et brukergrensesnitt og være i stand til å håndtere sertifikater for autentisering. Grensesnittet skal være oversiktlig og lett tilgjengelig for brukeren og kunne løse oppgavene deres på en effektiv måte. Ytterligere funksjonalitet er også en mulighet, dersom Goodtech ser nytten i det.

Teknologier:

Java - Programmeringsspråket API'et skal programmeres i.
https://www.java.com/en/

Vaadin - Rammeverket som skal brukes for å lage brukergrensesnittet.
https://vaadin.com/home

Eclipse Milo - Protokoll stacken API'et skal bruke.
https://projects.eclipse.org/projects/iot.milo

Maven - Et prosjektstyringsverktøy for programvare.
https://maven.apache.org/

Jenkins - Programvare for kontinuerlige integrasjoner og leveranser.
https://jenkins.io/

JUnit - Java rammeverk for testing.
http://junit.org/junit4/ eller
http://junit.org/junit5/

Spring - Java rammeverk for back-end for webapplikasjon.
https://spring.io/

Javadoc - Generering av dokumentasjon for API i HTML format.
http://www.oracle.com/technetwork/articles/java/index-jsp-135444.html

LaTeX - Redigeringsprogram for dokumenter.
http://www.latex-project.org/about/

IntelliJ - Utviklingsplattform API'et programmeres på.
https://www.jetbrains.com/idea/

freedcamp.com - Prosjektstyring, holder orden på oppgaver, frister, samt lagring av loggdokumenter o.l.
https://freedcamp.com/

Subversion - System for programvare versjonsstyring.
https://subversion.apache.org/

Google Docs - Gjør det mulig å dele tekstdokumenter.
https://docs.google.com

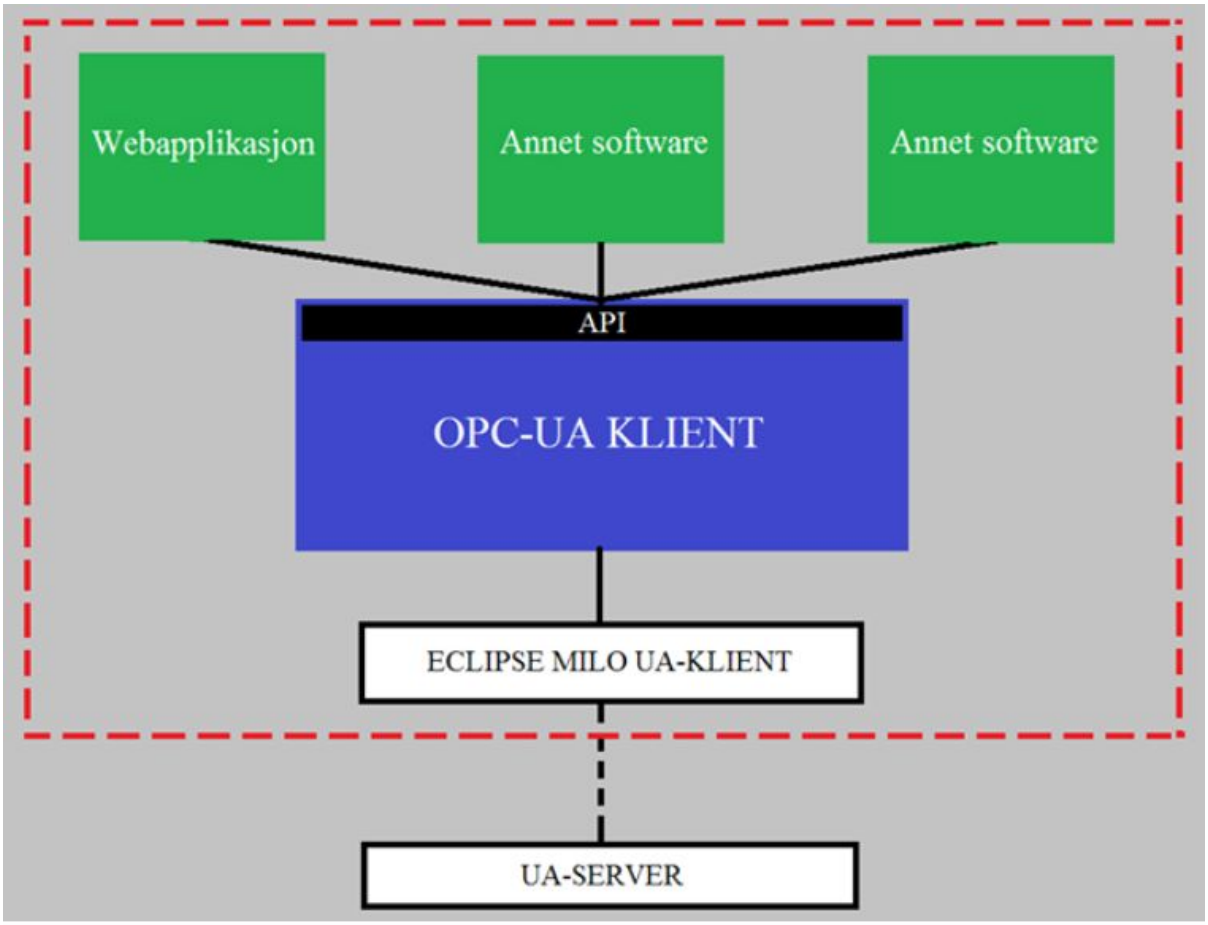
Facebook (Privat Facebookside og Messenger) - Kommunikasjon utenom "arbeidstid"
https://www.facebook.com/

Scrum - Arbeidsprosess, smidig utvikling, stand up, iterasjoner møter osv.
https://en.wikipedia.org/wiki/Scrum_(software_development) (info om scrum)

I tillegg ønsker gruppen å benytte seg av norsk til skriving av rapporten, mens programmeringen vil foregå på engelsk da dette er foretrukket av Goodtech. De har også gitt oss en liste med retningslinjer de bruker når det kommer til programmering som de vil at gruppen skal følge.

Løsninger/Alternativer

Løsningen vil gå ut på å programmere to moduler. Den ene modulen vil være en OPC-UA klient som fungerer som et API. Klientens hovedoppgave vil være å kommunisere med OPC-UA servere, i tillegg til å forsyne API'et med data. Den andre modulen vil være en webapplikasjon med et brukergrensesnitt, som bruker API'et til den førstnevnte modulen. Ved hjelp av rammeverkene Vaadin og Spring vil grensesnittet bli utviklet fra bunnen av, med utgangspunkt i originaldesignet. Dette er for at løsningen skal følge Goodtech sine retningslinjer, og passe med deres andre produkter.



Figuren viser webapplikasjonen og API'et denne skal bruke. Det er planlagt at alt innenfor den røde linjen skal kjøre på samme maskin.

Fordelen med å bytte ut det nåværende systemet med den nye løsningen er at den nye stacken som taes i bruk både er oppdatert og open-source, noe som gjør at kildekoden er gratis og tilgjengelig for alle. Det er også en fordel å bytte ut det gamle grensesnittet med noe forbedret, og ta for seg de delene av det gamle systemet som virket ineffektivt og mangelfullt. Siden gruppen allerede har tilgang til dagens versjon av systemet er det lett å finne forbedringer og se muligheter til å effektivisere og tilgjengeliggjøre grensesnittet enda mer. En ulempe med løsningen er at utviklerne hos Goodtech må lære seg det nye APIet, for å kunne bruke det i fremtidige prosjekter. Goodtech har bestemt hvilke teknologier som skal brukes i oppgaven, men gruppen står fritt til å velge løsning ut i fra de teknologiene som er valgt.

Analyse av virkninger

Oppgaven er av typen der det i liten grad er mulighet for å velge løsninger. Hvordan gruppen bruker teknologiene til å lage løsninger kommer til å bli drøftet senere i prosjektet, da dette blir for spesifikt for denne rapporten.

Ordforklaringer

[1] API	**Application Programming Interface** Et sett med kommunikasjonsmetoder mellom software komponenter. Typisk en byggestein for en applikasjon.
[2] Stack	I denne sammenhengen menes protokoll stack. En protokoll stack er en samling av nettverksprotokoller som jobber sammen, stablet ("stacket") oppe på hverandre.
[3] OPC-UA	**Open Platform Communications - Unified Architecture** En protokoll for dataoverføring ofte brukt i industriell sammenheng. Laget av OPC foundation: https://opcfoundation.org/ Mer om OPC-UA: https://opcfoundation.org/about/opc-technologies/opc-ua/
[4] Teknisk gjeld	Er definert som alt det uferdige, unødvendig kompliserte og utdaterte i løsningene som hindrer bedrifter i å drifte, forvalte og videreutvikle så effektivt som bedriftene burde.

Risikoanalyse

Risikoanalysen viser hvilke risiko som kan forekomme i ulike situasjoner av det gruppen tenker er viktig å ta for seg. Den går fra lav til veldig høy der lav er det minst kritiske og veldig høy er det mest kritiske. I forebygging har det blitt skrevet en tekst som beskriver konsekvensene av situasjonen og hva som kan gjøres for å forebygge situasjonene.

Situasjon	Sannsynlighet	Konsekvens	Forebygging
Tap av data	Lav	Veldig høy	Tap av data vil føre til store konsekvenser ved at prosjektet ikke blir ferdig eller at gruppen får et stort tilbakefall. Generell forebygging av tap av data er å ta backup av data. For dette prosjektet blir det brukt Subversion til å lagre kildekode og dokumentasjon på Goodtech sine servere, i tillegg til diverse lokal backup.
Sykdom	Høy	Middels	Kortvarig og langvarig sykdom kan føre til at utsettelser forekommer. Forebygging vil da være at alle har en oversikt over hva alle gjør slik at andre kan ta over.
Feilberegne tid	Middels	Middels	Feilberegning av tid kan forekomme hvis gruppen bruker for mye tid på hver enkelt oppgave og det vil ha konsekvenser hvis en er bak skjema. Forebygging er å prøve å følge planen som lages i startfasen og begrense oppgaven ved å prioritere de viktige deloppgavene først.
Ikke ferdig prosjekt	Lav	Middels	Oppgavens omfang kan bli større og mer omfattende i løpet av den gitte tiden enn hva gruppen hadde sett for seg. Forebygging vil være å legge opp til videreutvikling slik at oppdragsgiver kan eventuelt fortsette.
Umulig å gjennomføre	Lav	Middels	I løpet av prosjektet kan det hende det oppstår problemer som ikke lar seg løse av gruppen. Et eksempel på et slikt problem er hvis det oppstår en feil med OPC-UA implementasjonen Eclipse Milo. Da denne er relativt ny kan det være mangler eller feil, og det kan få konsekvenser for funksjonaliteten til løsningen.Hvis det oppstår et problem med Eclipse Milo vil det være hensiktsmessig å rapportere feilen til stackens utvikler.

Fremdriftsplan

Fremdriftsplanen viser hvordan gruppen ser for seg at tiden skal bli brukt og varighet.

Fase	Varighet	Beskrivelse
Oppstartsfase	02.12.16 til 05.01.17	I denne fasen skal gruppen ha et møte med arbeidsgiver, holde gruppemøter og skrive prosjektskisse.
Forprosjekt	10.01 til 20.01.17	Målet for denne fasen er å lage en presentasjon med mål og rammebetingelser, løsninger/alternativer, teknologi, utforme fremdriftsplan, risikoanalyse og lage et sammendrag. Videre skal gruppen ha et møte med oppdragsgiver og møte med veileder. Gruppen starter å bruke oppdragsgivers lokaler. Nettside for prosjektet skal opprettes i denne fasen. Gruppen starter å lære om OPC-UA og får en oversikt over den gamle, fungerende løsningen som skal forbedres.
Startfase	23.01 til 06.02.17	I startfasen skal gruppen få en større oversikt over hovedoppgaven. Det innebærer å bli bedre kjent med den gamle løsningen og teknologien vi skal bruke. Videre skal gruppen begynne planleggingen for fullt ved å fordele oppgaver som må bli gjort med modulene som skal utvikles.
Utviklingsfase	08.02 til 28.04.17	Utviklingsfasen er den største fasen. Gruppen starter med utviklingen av modulene, parallelt med diverse testing. Første del av fasen vil gå ut på å lage et proof-of-concept, som vil bestå av “ryggraden” til hele systemet, for å minimere risikoen for å møte problemer senere i prosjektet. I tillegg skal det også produseres prosjektdokumentasjon. På slutten av fasen skal resultatet være ferdig nok til at det kan bli utsatt for mer omfattende testing i neste fase.
Testfase	01.05 til 12.05.17	Testing av hele systemet er målet for denne fasen. Gruppen kommer til å gjøre det ved hjelp av enhetstesting, brukertesting, systemtesting, interface testing og sikkerhetstesting. Arbeidsgiver kommer også til å utføre omfattende testing før de vurderer om de skal implementere løsningen i sitt system, men det er ikke en del av denne fasen.
Sluttfase	15.05 til 24.05.17	Gruppen skal se gjennom kode og dokumentasjon. Til slutt skal oppgaven leveres.
Presentasjon	06.06 til 09.06.17	Muntlig presentasjon av oppgaven.

Kilder

Bildet er hentet fra: http://www.ni.com/cms/images/devzone/tut/Figure_1_20120304162228.jpg
(Beskjært)