Luleå University of Technology

Third year project

# Sensor data aggregation through CoAP

**Authors:**
Sophia Bergendahl
*sopber-8@student.ltu.se*

Edvin Bruun
*edvbru-9@student.ltu.se*

William Gustafsson
*wilgus-9@student.ltu.se*

Christoffer Holmstedt
*cihhol-7@student.ltu.se*

Marcus Rådman
*marrdm-9@student.ltu.se*

Kristoffer Svensson
*kirsev-9@student.ltu.se*

Ludwig Thurfjell
*ludthu-7@student.ltu.se*

**Supervisors:**
Ulf Bodin
*ulf.bodin@ltu.se*

Jens Eliasson
*jens.eliasson@ltu.se*

Rumen Kyusakov
*rumen.kyusakov@ltu.se*

February 28, 2012

# Project Description

## Background

Luleå University of Technology conducts research on lowpower wireless microprocessors called "Mulle". These microprocessors can be used for various things depending on which type of sensors you connect to it, everything from measuring temperature or vibrations in a car to analyzing the quality of the road that you drive on.

Every year northern parts of Sweden are used for testing cars during winter conditions. To test a car you first decide what you want to test, then you test with local sensors logging within the car. When enough data is collected you return back home. At the testing facility the data is now available for analysis. Depending on the results from the previous runs you might want to test some parts in more detail so you re-configure all sensors and go out for another test run.

This process is time consuming when you need to return to testing facility to be able to analyze and re-configure all sensors. In todays society most computers are connected to internet and/or other private networks, most of these computers have the ability to be remotely configured and maintained. The goal with this project is to be able to analyze data from sensors in realtime and re-configure them on the fly while testing is in progress.

## Project Targets

1. Be able to send live sensor data from multiple "Mulles" to an online logging server/service.

2. Be able to read sensor data on the web with both a PC (web browser) and through an Android mobile device.

3. Be able to re-configure the sensors through a web interface and through an Android mobile device.

## Technical dilimiations

TODO: Vad har explicit uteslutits från arbetet?

# Execution of the project

## Scrum and how it has been used

It was decided back in november that the entire project would be divded into three sprints. The exact dates were to be decided in the beginning of each sprint. In cooperation with the client the scope of the project and the scope of the first sprint was decided upon in november. During the first projectmeeting the first sprint goal was divided into eight sprint stories. It soon became clear that those eight stories were way to big, at the end of the sprint none of the stories had been finished.

Lesson learnt, the second sprint was divided into smaller stories which gave immidiate result when the first 69 sprint story points finished during the second sprint.

To decide upon size for each sprint story, for the second and third sprint, ”planning poker” [1, p. 42] was used. For every sprint story each project member wrote down an estimate on the scope for each story. With planning poker it became clear that each project member had a different vision for each story. A short discussion after each estimate made it more clear on how big the scope was, an agreement was usually made within a few minutes.

## One project, three sprint goals

As mentioned earlier the project was divided into three sprints. This meant that three different sprint goals had to be divided into smaller sprint stories which in turn had to be assigned to a project member. Due to all project members being new to most of the tasks at hand the first team division was made with focus on components [1, p. 106]. The goal with this was that each smaller team within the project could sit together and dive deep into their specific part such as the Mulle, the server parts or the android code. Later on a split into cross-component teams [1, p. 107] was aimed for but due to some persistent bottlenecks in some components this was never done.

For all sprints the sprint planning meeting were used to categorize each sprint story into the different components (Mulle, Server, Android). It was then up to each component based team to split their stories between themselves. This ended being a very flexible solution, in some cases, to flexible when a team didn’t use the Scrumboard online at Scrumdo.com the team could wander off from the sprint story they were supposed to work on. For the last sprint everyone got at least one sprint story assigned to themselves directly during the sprint planning meeting. This was made to put more focus on using the Scrumboard at Scrumdo.com.

A move to cross-component teams was never made instead an attempt to increase speed for the Mulle and the Android component was made by moving one team member from the server team to the Mulle team and another to the Android team. The server component at this time was way ahead of the other components. Another week later it became clear that the additional team member for Mulle team was not needed, cause the problem was still a bottleneck that only one or two team members could work on at a single point in time, a move back to the server team was made.

## Individual time monitoring and speed

### Sophia Bergendahl

### Edvin Bruun

### William Gustafsson

### Christoffer Holmstedt

Exempelvis kan lista användas om man vill. Jag, christoffer, kommer inte använda det tror jag.

1. TODO: Hur man citerar till specifik sida i kurslitteraturen. [1, p. 42]
2. TODO: Hur man citerar utan sidhänvisning [1]
3. Third sprint story

### Marcus Rådman

### Kristoffer Svensson

### Ludwig Thurfjell

## Reflection about Scrum usage during this project

# Results

## Deliverables

TODO: Vad levererar vi med respekt till ursprungliga krav?

## Testing

During this project testing has been made by each project member. The scope of the testing for each story has been up to each project member to decide upon. Right before each sprint demo a project meeting was scheduled were each one showed what was completed and what was not. Depending on what was ready at that point in time the entire Scrum team decided what was going to be presented at the demo and a test was made to confirm that it was possible to show the parts that were decided upon. No further testing was made during this project.

## Lessons learnt

TODO: Jobba inte på så många olika delar samtidigt. Välj ut tydliga komponenter man jobbar på först sedan tar man andra när man känner att möjligheten finns. TODO: Skapa tester specifikt för varje komponent och inte ”kommunikationen fungerar emellan dessa två komponenter” enbart. Med enbart det senare alternativet kommer en färdig komponent inte kunna testas utan att den andra är klar vilket skapar en flaskhals. TODO: Det går aldrig att skapa för detaljerade tester, se till att varje mål har en tydligt definition på när den är klar ”How to demo”.

## Suggested improvements

# Conclusions

# References

[1] Henrik Kniberg, *Scrum and XP from the Trenches.* C4Media Inc, Publisher of InfoQ.com, 978-1-4303-2264-1, http://infoq.com/minibooks/scrum-xp-from-the-trenches, 2007.

# Appendix A - How to build upon our codebase

This appendix include information on how to build upon our codebase for the Mulle (C), server code (Python, PHP/HTML5 and C) and Android Mobile phone (Java).

## Mulle

## Server

### Coapy server

TODO: Python parts such as the python coapy server and how we use EXIP c-code parts.

### Webpages and database

## Android Mobile Phone application