

Project handbook

Contents

<i>Work agreement</i>	2
1.1 Roller og oppgavefordeling	2
1.2 Prosedyrer	2
1.3 Interaksjon.....	3
1.4 Progress plan	4
<i>Meeting notices</i>	5
<i>Meeting notes</i>	18
<i>Sprint reports</i>	29
<i>Sprint retrospectives</i>	37
<i>Weekly reports</i>	45
<i>Delegations of responsibility</i>	62
1.5 Backend and operations	62
1.6 Frontend.....	62
<i>Timesheets</i>	63
David	63
Einar	68

Work agreement

Arbeidskontrakt for Time-series cache

Medlemmer: Einar, David

Einar Aglen
.....

David Rie Knotten
.....

Innledende tekst

Denne arbeidskontrakten bygger på et sett med typiske mål, oppgavefordelinger, prosedyrer og retningslinjer for interaksjoner for studentarbeider. Arbeidskontrakten er utfylt med egne fortolkninger av hva man mener med disse og hvordan man skal oppnå dette.

1.1 Roller og oppgavefordeling

Frontend, Einar: Ansvarlig for at brukergrensesnitt skal kunne brukes av alle, og at produktet skal være på standard med hva som kan forventes i dagens industri

Backend, David: Står som hovedansvarlig for at backenden er opp til standard og har en leveringsklar kvalitet.

Dokumentansvarlig, David, Einar: Begge gruppemedlemmene har ett likestilt ansvar for å oppretholde god dokumentasjon

Dersom arbeidsmengdene varierer stort må begge gruppemedlemmene kunne bidra på andres hovedoppgaver

1.2 Prosedyrer

A. Møteinnkalling

Hver ukedag skal det tilpasses ett standupmøte til en tid som passer alle medlem. Dette avtales gjennom dialog mellom gruppemedlemmene.

Det skal og bli holdt ett møte med veileder og oppgavegiver. Møtetid for neste møte blir avtalt under hvert møte. Det første møtet må det bare finnest en tid hvor det passer for alle

Dersom det trengs møter utenom dette må møtene bli avklart mellom partene som skal/må delta

B. Varsling ved fravær eller andre hendelser

Dersom man kommer for sent eller ikke kan møte, må en varsle fra i god tid og med en rimelig grunn

C. Dokumenthåndtering

For versjonsstyring av kode bruker gruppen GitHub.

For annen dokumentasjon skal gruppen bruke Confluence, for wiki og filer, og Jira for issues og generelt prosjekt planlegging

D. Innleveringer av gruppearbeider

Ett hvert gruppemedlem skal ha sitt arbeid klargjort til innleveringer i rimelig tid slik at andre gruppemedlemmer skal kunne se over stoffet og kvaliteten.

For kode blir det ikke stilt særlige krav utenom fornuftig dokumentasjon for koden slik at den er forståelig for de andre medlemmene.

1.3 Interaksjon

A. Oppmøte og forberedelse

Gruppemedlemene skal møte opp til avtalt tid, med unntak dersom der er skjærskilte grunnlag til forsentkommelse. Ved standup møter er det en forventning av at et hvert medlem skal ha en viss oversikt over hva som har blitt gjort. Ved møter med veileder og/eller oppgavegiver bør gruppen ha laget en felles oversikt over hva som er blitt gjort eller ikke.

B. Tilstedeværelse og engasjement

Under møter skal gruppemedlemmenes fokus være møtet. Så vidt det lar seg gjøre skal medlemmene unngå avbrytelser / forstyrrelser fra møtene.

C. Hvordan støtte hverandre

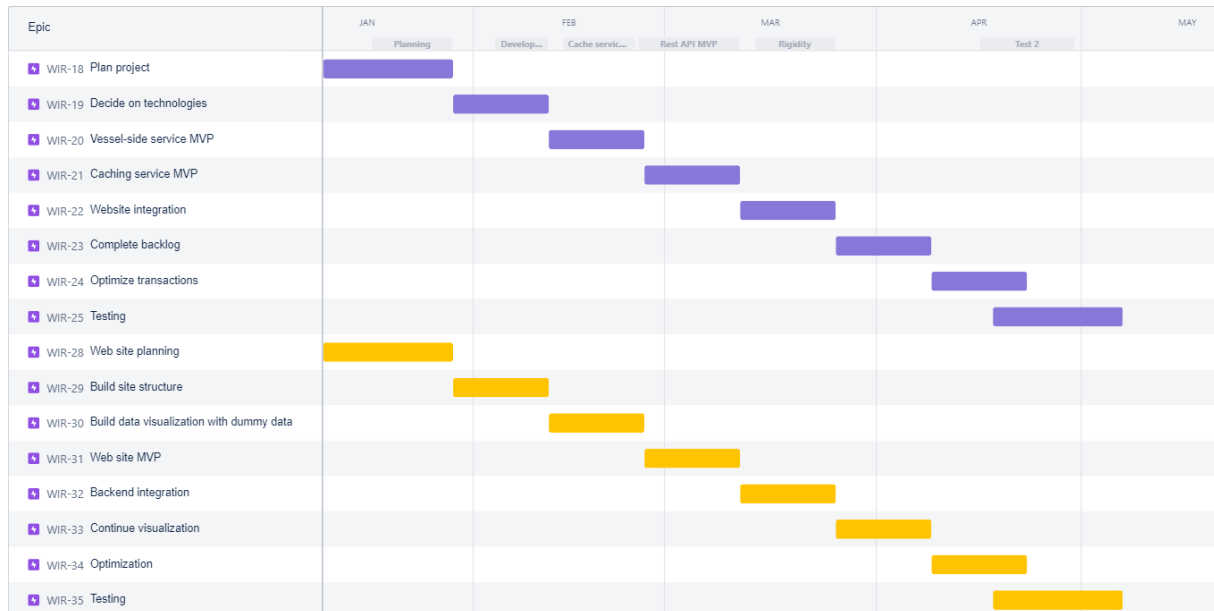
Hvert gruppemedlem skal stille seg disponible for resten av gruppen under normale arbeidstimer.

D. Uenighet, avtalebrudd

Håndteringen av uenigheter og avtalebrudd skal snarest oppløses innad i gruppen. Dersom gruppen ikke kommer til enighet skal veileder eller eventuelt oppgavegiver involveres for å megle frem en løsning.

Hvilke avvik som er akseptert må bli løst på ett rimelig vis og av hele gruppen

1.4 Progress plan




Sprints have been bi-weekly with the following sprint being planned after the end-of-sprint meeting with Girts and Seaonics.


The periods on top of the graph are the actual sprints that the group performed, with the diagram being the planned sprints.

While there have been some differences between the plan and the project the plan was followed to some degree. The main differences have been that the Vessel-side service was dropped from the project, thus putting all backend sprints one sprint ahead. During the rigidity sprint, the group worked on making the service more rigid as it had a lot of issues and edge cases. The following sprints, test and test 2 (The first test sprint has for some reason disappeared), were used to test the service through user tests, but also torture tests. This took a much longer time than anticipated, since more and more errors seemed to occur and new sub-systems were developed to deal with them, for example the asynchronous streaming system.


Meeting notices

 **Bachelor oppgave - Seaonics** Chat Files Details

✓ Accepted ▾ | ⇄ Copy link → Forward Show as: Busy ▾ Category: None ▾



 Friday 14. January 2022 14:00 - 15:00 [Show meeting info](#) ▾



 Hei Einar

På tirsdag tar vi en kort gjennomgang internt med hensyn til oppgave, om vi skal jobbe rundt det vi har snakket om tidligere eller om der er andre løp dere kan ta.

Kan du og Daniel sende meg bare noe linjer på hva dere ønsker å jobbe med som teknisk utfordring?

Microsoft Teams meeting

Join on your computer or mobile app

[Click here to join the meeting](#)

[Learn More | Meeting options](#)



Bachelor oppgave Seaonics

[Chat](#)[Files](#)[Details](#)

✓ Accepted ▾

↔ Copy link

→ Forward

Show as: Busy ▾

Category: None ▾



Friday 28. January 2022 13:00 - 14:00

[Show meeting info](#) ▾



Takk for møte I dag.

Sender som avtalt nytt møtetidspunkt om 14.dager

Microsoft Teams meeting

Join on your computer or mobile app

[Click here to join the meeting](#)

[Learn More](#) | [Meeting options](#)



Bachelor oppgave, Remote Diagnostics and Data collection

[Chat](#)[Files](#)[Details](#)

✓ Accepted ▾

↔ Copy link

→ Forward

Show as: Busy ▾

Category: None ▾



Wednesday 9. February 2022 16:00 - 17:00

[Show meeting info](#) ▾



Microsoft Teams meeting

Join on your computer or mobile app

[Click here to join the meeting](#)

[Learn more](#) | [Meeting options](#)



Bachelor Gruppe Seabrief Caching

[Chat](#)[Files](#)[Details](#)[← RSVP](#) ▾[↔ Copy link](#)[→ Forward](#)[Show as: Tentative](#) ▾[Category: None](#) ▾

Thursday 3. March 2022 13:00 - 14:00

[Show meeting info](#) ▾

Microsoft Teams meeting

Join on your computer or mobile app

[Click here to join the meeting](#)

[Learn more](#) | [Meeting options](#)



Wirelogger internal meeting

[Chat](#)[Files](#)[Details](#)[Meeting notes](#)[Whiteboard](#)

✓ Accepted ▾

↻ Copy link

→ Forward

Show as: Busy ▾

Category: None ▾



Tuesday 22. March 2022 09:00 - 10:00

[Show meeting info](#) ▾



Microsoft Teams meeting

Join on your computer or mobile app

[Click here to join the meeting](#)

Or call in (audio only)

+47 21 40 22 36,,170495100# Norway, Oslo

Phone Conference ID: 170 495 100#

[Find a local number](#) | [Reset PIN](#)

[Learn More](#) | [Meeting options](#)

Wirelogger bachelor projekt Chat Files **Details** Scheduling Assistant Meeting notes Whiteboard Attendance Breakout rooms +

Cancel meeting Copy link Show as: Busy Category: None Time zone: (UTC+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna Meeting options Response options Require registration: None

Monday 28. March 2022 12:00 - 12:30 View series Show meeting info

Attendance


B I U S V A Paragraph Paragraph | Bulleted list Numbered list Link Unlink Undo Redo

Foreslått ending til neste møte


Microsoft Teams meeting


Join on your computer or mobile app
[Click here to join the meeting](#)

[Learn More](#) | [Meeting options](#)

 **Brukertesting av Remote diagnostikk / Signal grafing** Chat Files **Details** Recordings & Transcripts


[← RSVP](#) | [↔ Copy link](#) [→ Forward](#) Show as: Tentative [Category: None](#)

 Thursday 21. April 2022 12:30 - 14:00 [Show meeting info](#)



Recording

Multiple available



Brukertesting av Remote signal grafing / Caching tjeneste

Microsoft Teams meeting

Join on your computer or mobile app
[Click here to join the meeting](#)

[Learn More](#) | [Meeting options](#)

Meeting notes

2022-01-20 Meeting notes

 Date

14 Jan 2022

 Participants

[David Rise Knotten](#)

[einar](#)

[Girts Strazdins](#)

[Espenakker](#)

[Stig Espeseth](#)

 Goals


Gather information on the problem at hand


Requirements

Pre-existing systems

 Discussion topics

Time	Item	Presenter	Notes
1hr	Orientation	Everyone	What's the main goal of the task Contracts Obligations

 Action items

 Decisions

2022-01-28 Meeting notes



Date

28 Jan 2022



Participants

[David Rise Knotten](#)

[einar](#)

[Girts Strazdins](#)

[Espenakker](#)

[Stig Espeseth](#)



Goals

Get everyone on the same plane in terms of the plan for the project

Finish up the contracts



Discussion topics

Time	Item	Presenter	Notes
30m	Discuss the project and plan	Everyone	
10m	Figure out the contracts	Everyone	



Action items

Finish up contracts and hand in pre-project plan

[David Rise Knotten](#) Set up meetings going forward



Decisions

2022-02-11 Meeting notes



Date

11 Feb 2022



Participants

[David Rise Knotten](#)

[einar](#)

[Girts Strazdins](#)

[Espenakker](#)

[Stig Espeseth](#)



Goals

Discuss further progress



Discussion topics

Time	Item	Presenter	Notes
15m	Retrospective	David	
15m	Security	Everyone	The groups should discuss implications in the report A simple access management system is wanted Data used for presentations or on public domain should be abstracted
5m	The road ahead	Everyone	The groups main goal for the coming sprint is to create a functioning prototype of the caching service



Action items

Write about security options

Start development for Caching Service MVP



Decisions

2022-02-25 Meeting notes

 Date

25 Feb 2022

 Participants

[David Rise Knotten](#)

[einar](#)

[Girts Strazdins](#)

[Espenakker](#)

[Stig Espeseth](#)

 Goals


Get feedback on web app

What can improve on current solution

Generally: The road forward

 Discussion topics

Time	Item	Presenter	Notes
11:30	Status	David Rise Knotten einar	Show website Discuss issues
11:40	Data savings	Tom	
11:50	Statistics	Tom & Stig	
12:00	Signal and graph manipulation	Tom	
12:10	Issues	einar David Rise Knotten	

 Action items

 Decisions

2022-03-02 Meeting notes

 Date

02 Mar 2022

 Participants

[David Rise Knotten](#)

[einar](#)

[Espenakker](#)

[Stig Espeseth](#)

[Girts Strazdins](#)


 Goals

Discuss further development and features

Feedback on current functionality

 Discussion topics

Time	Item	Presenter	Notes
60m	Testing and discussing data hydration of chart	Everyone	
20m	Improving endpoints and possibility for vessel service	Everyone	

 Action items

Work on data hydration based on chart navigation

Improve endpoints

Work on WebSocket solution

 Decisions

2022-03-11 Meeting notes

 Date

11 Mar 2022

 Participants

[David Rise Knotten](#)

[einar](#)

[Espenakker](#)

[Girts Strazdins](#)

[Stig Espeseth](#)

 Goals


Detailed use-cases for improving work-flow of website

Discuss additional features

Thoughts about current version

 Discussion topics

Time	Item	Presenter	Notes
5 min	Reschedule next week	David Rise Knotten	Monday week after 12:00-12:30
15 min	Demonstrate website	einar	
5 min	Longer intervals	Espenakker	Seeing a whole day summarized by a subset of ranges
5 min	avg + min/max	Stig Espeseth	
10 min	Improvements		Axe control, loading indicator

 Action items

Reschedule next meeting

 Decisions

2022-03-22 Meeting notes

 Date

22 Mar 2022

 Participants

[David Rise Knotten](#)

[einar](#)

[Girts Strazdins](#)

 Goals

Discuss the main Report

Structure

Inspirations

Content/Topics

Pin point important topics

Cut out unnecessary parts

Discuss reports


Review


Retrospective

Weekly reports

 Discussion topics

Time	Item	Presenter	Notes
30 min	Main report	Everyone	Talk a bit about progression
30 min	Additions to report	Everyone	What more can be added

 Action items

 Decisions

2022-03-28 Meeting notes



Date

28 Mar 2022



Participants

[David Rise Knotten](#)

[einar](#)

[Espenakker](#)

[Girts Strazdins](#)

[Stig Espeseth](#)



Goals

Show changes

Discuss testing



Discussion topics

Time	Item	Presenter	Notes
5 min	Review	David Rise Knotten	
10 min	testing	David Rise Knotten	
5 min	Demo	David Rise Knotten	
5 min	Vessel discovery	einar	
5 min	Sparkplug b	Stig Espeseth	



Action items



Decisions

2022-04-08 Meeting notes



Date

08 Apr 2022



Participants

[David Rise Knotten](#)

[einar](#)

[Girts Strazdins](#)

[Espenakker](#)

[Stig Espeseth](#)



Goals



Discussion topics

Time	Item	Presenter	Notes
5 min	status/progress	David Rise Knotten	
25min	Website demo	Espenakker	
5min	Road forward		
15min	Discussing report		



Action items



Decisions

2022-04-22 Meeting notes



Date

22 Apr 2022



Participants

[einar](#)

[David Rise Knotten](#)

[Stig Espeseth](#)

[Espenakker](#)



Goals

Present results from current sprint

Present results from user testing

Talk about last part of development



Discussion topics

Time	Item	Presenter	Notes
10 min	Backend improvements	David Rise Knotten	Discussed how improvements has impacted the projects
10 min	Website improvements	einar	Discussed new features and changes
10 min	User Testing	all	Discussed comments from testing, and future improvements



Action items



Decisions

2022-05-06 Meeting notes



Date

06 May 2022



Participants

[David Rise Knotten](#)

[einar](#)

[Girts Strazdins](#)

[Espenakker](#)

[Stig Espeseth](#)



Goals

Discuss progress



Discussion topics

Time	Item	Presenter	Notes
30min	discussing progress		



Action items



Decisions

Sprint reports

Planning sprint review

Sprint goals

During the planning sprint the main goal is to formulate a plan and a scope for the project

Planned activities

- Set up development tools
- Set up documentation tools
- Write preliminary report
- Define issue
- Define main goals
- Define scope
- Define project organization
- Create an estimate for the project plan
- Meet with Seaonics to discuss issue
- Create and sign work agreement
- Sign three part agreement

Actually performed activities

- Completion on work agreement
- Talked with Seaonics and set expectations
- Completed preliminary report
- Created estimates
- Set up documentation and development tools

Deviations from plan and why

The group failed to get the three part agreement signed during the first sprint due to some decisions having to be made on behalf of Seaonics as well as a somewhat late plan. The definition of issues, goals and scope were also left somewhat fuzzy.

Plan forward

After the group has finished creating a plan, the coming sprints main goal will be to set up environments for testing and all dependencies the project has, so that the group is ready to start developing

Development start review

Sprint goals

During the development sprint the goal was to start to actually build some examples and test programs for the project, getting a feel for what kinds of tools the solution should contain.

Planned activities

- Decide on technologies
- Languages
- Frameworks
- Database
- Figure out how data will be structured
- Figure out website structure
- Create architectural model of the overall system

Actually performed activities

- Created an architectural model for the services that will run in the cloud
- Started working on the cache service
- Development started on Node using TypeScript
- No framework is used for the cache service
- Set up database
- Set up MQTT broker
- Created data model and system architecture model
- Created first version of wireframes
- Created basic structure of website
- Website created using Next.js and Chakra UI
- Learning to use SSR

Deviations from plan and why

The workload during this sprint was a bit too light for two weeks, and all planned goals were met

Plan forward

The plan going forward will be to start finish the MVP of the cache service, as well as to start working on the website and if time allows, the web-facing REST service

Cache service review

Sprint goals

The goal of this sprint is to complete an MVP of the cache service and have it provide an API for other MQTT clients.

Planned activities

- Create fetch last functionality on cache service
- Create fetch range functionality on cache service
- Cache data and combine data to avoid transferring the same data more than necessary
- Unit test server logic
- Start setting up data visualization tools for handling the cached data
- Create Tools for handling network requests to the REST API
- Implement data types for handling cached data on website
- Start work on rest service and get the services to communicate

Actually performed activities

- Created fetch functionality for last and range
- Implemented caching with lazy loading, requesting data from the vessels when queries cannot be fulfilled with cached data.
- Connect website with REST API
- Use fetched data in data visualization
- Early implementation of WebSocket
- Written tests covering most of the server logic
- Started working on the REST service
- Started work on REST service, even creating web-facing interfaces
- Deployed MVPs on AWS

Deviations from plan and why

The past two weeks can be described as highly efficient and even though the workload has been somewhat sizeable, all set goals have been completed.

Plan forward

The plan going forward will be to implement features requested in the last meeting with Seaonics, as well as expanding the REST API to do more data handling.

Rest API MVP review

Sprint goals

The goal of this sprint is to end up with a finished REST service, improve data visualization and start working on metadata on the cache service

Planned activities

- Enable TLS for MQTT
- Enable TLS for Postgres
- Define MetaData
- Create websocket stream on REST service for contiguous data streaming
- Logging of incoming and outgoing data
- Create system for tracking data usage per session
- Use metadata to know when to fetch new data
- Find way to add y-axis for each dataset
- Change x-axis to fit data
- Find way to fetch data on-zoom and on-pan

Actually performed activities

- TLS for both MQTT and Postgres
- Websocket on REST service
- Outgoing and incoming data logged
- Metadata defined
- Axis improvements added
- Data fetching when navigating the chart

Deviations from plan and why

The group shifted focus during the sprint from development over to documentation, having spent most of the time working on the project report. Therefore some planned activities were left incomplete.

Metadata is not yet used by the cache to decide when to fetch new data from the vessel

While I/O of data is logged, it's not tied to a session, nor is it used in any meaningful way, yet

Re-working work-flow, needs use-cases before starting

Plan forward

Going forward, the group will have a focus on both the documentation and reports, as well as further development. For the backend, the development focus will be on logging and the use of metadata to avoid double-fetching the same data multiple times

Rigidity sprint review

Sprint goals

The main focus of this sprint is to improve the general rigidity of the system and improving the “behind the scenes”. Currently, the system has the wanted functionality, but there are quite a few breaking bugs that need to be eliminated

Planned activities

- Fix all bugs related to MQTT connectivity
- Make use of MetaData to avoid double-fetching
- Change time representation to be unix time
- Use I/O logs to track the data savings of the service
- Continue improving documentation
- Report the over-wire sizes of data transfer on requests

Actually performed activities

- Report bytecost of service usage
- Use unix time instead of ISO date strings
- Metadata made use of
- I/O logging, including over-the wire transfer sizes for requests
- Some more documentation
- Most, if not all, MQTT “breaking” bugs fixed

Deviations from plan and why

No progress for frontend as a result of exam preparations and work weekend

Plan forward

The plan forward is to start testing the system and diving deeper into documentation,

Testing Sprint Review

Sprint goals

The goals for this sprint are to perform user tests on the system as well as smooth out issues found

Planned activities

- Set up environment for user testing
- User testing
- Display data usage in frontend
- Add colors to chart scales
- Get feedback on HMI functionality

Actually performed activities

- Set up user testing environment
- Some user-testing, but not enough
- Displayed data usage in frontend
- Added scale colors matching chart-lines for website
- Some bugfixing from limited user testing
- Decent amount of documentation

Deviations from plan and why

User testing, Took some time getting into the office and setting up user environment, where multiple new bugs were found. Had to postpone user testing

Did not receive as “pure” of a feedback as we had hoped, since user testing was only done by technical personell.

Plan forward

Going forward we'll have to perform some more user testing where different people try to make use of the system in a simulated environment.

Testing 2 Sprint Review

Sprint goals

The goals for this sprint are to perform user tests on the system as well as smooth out issues found

Planned activities

- User-testing
- Fetch interval splitting
- Migration to copy-stream instead of SQL inserts
- System to keep control of the state of currently streaming data
- API rework
- Add error messages to API
- Make all communication from/to cache protobuf
- Healthcheck endpoints
- Database rework
- Synchronous metadata aggregator (versus constant background task)
- Patch current-time edge-case
- Update REST service to support the above

Actually performed activities

- User-testing
- Fetching split into 50second intervals
- Database inserts replaced by streams
- streamState object created
- API reworked
- Everything through protobuf
- Optional error messages added
- Healthcheck system added
- Database rework
- Data split into tables, saving 56 bytes/record
- Metadata aggregator now running synchronously
- Patched current-time edge-case
- REST fully supporting all added functionality

Deviations from plan and why

No deviations

Plan forward

The plan forward is to add a simple command API to clear out data from the cache, as its size may become a problem over time

Rigidity 2 Sprint Review

Sprint goals

- Further improve the rigidity of the cache
- Planned activities
- Responsive intervals
- Command API
 - Wipe command
- Metadata API
- Hashing names
- Improve tests
- Stop healthcheck polling, and make it interval based
- Run healthchecks centralized in separate processes to avoid issues with cpu-intensive loads
- Eliminate all dependencies from tests
- Improve Error handling for Website
- Refactor MQTT getters to use a client-pool
- Improve JSDoc

Actually performed activities

- Made fetch-intervals respond to how quick responses are
- Added endpoints for fetching metadata
- Added simple command api where data can be deleted
- Names hashed in the database
- Tests improved as well as coverage
- Added stream-state controller
- Central controller keeping tabs on all current streams
- Added entity-data controller
- Central controller keeping tabs on all entities (vessels and variables), making all their information available.
- Healthchecks now interval based
- Cache “pongs” every 5 seconds, REST api checks for pongs every 6 seconds
- Healthchecks running as separate processes on cache, and is centralized on REST
- REST service now using observer pattern for individual requests to be notified if healthchecks fail
- Now using in-memory database for unit tests
- Added Debounce / Rate Limit to arrow-key movement for Website to reduce requests
- JSDocs for all repositories
- Added more detailed messages in feedback to user when fetching new chart data
- MQTT clients pooled in both services

Deviations from plan and why

- Wipe command incomplete, does not handle MetaData
 - Postponed due to high workload

Plan forward

The plan forward will be to finish up the bachelors report and complete the shortcomings of this sprint, as well as fixing any more bugs if discovered

Sprint retrospectives

Planning sprint

Overview

Over the course of the planning sprint the group has looked through different technologies, their upsides and downsides. There has also been some discussion/meetings to gain as much information about the issue the project should solve and its requirements

Date	27.01.2022
Team	Wirelogger
Participants	David Rise Knotten einar

Retrospective

Start doing	Stop doing	Keep doing
Breaking down issues into smaller issues		Documenting processes
Standup meetings		Good teamwork

Action items

- Standup atleast 2 times a week

Development start



Overview

The past two weeks of development have been quite slow due to a lot of work on other courses. The team has set up some basic software for the caching service with database and mqtt clients. During a meeting with Eirik the team went through the communication protocol that is to be used, giving a clearer picture of what the caching service will look like. The end result of the sprint is all the different components needed to build the system ready, but not yet interoperable

Date	10 Feb 2022
Team	Wirelogger
Participants	David Rise Knotten einar



Retrospective

Start doing	Stop doing	Keep doing
Setting up clearer, simpler tasks and better planning in general	Spending too much time at work	Documenting as we go
Pair programming		



Action items

Cache Service Sprint

Overview

Over the past two weeks the group has worked hard developing the cache service, REST service and website. Currently the cache service is mostly done, with a few semantic issues and a lack of user testing. It does, however have a solid unit test coverage.

The rest service has also been built, and acts as a web-facing interface of the cache service.

The website now has the ability to display data types dynamically through the REST services discover function, and a couple hardcoded vessels. The plan further along the road is to create a discover function for vessels as well mirroring the signal discovery.

All resources are deployed and available for testing on <https://wirelogger.com/>. Although keep in mind the server it runs on is fairly weak and pushing limits with ranges and or data points is likely to crash it.

Date	24 Feb 2022
Team	Wirelogger
Participants	David Rise Knotten einar

Retrospective

Start doing	Stop doing	Keep doing
Documenting		Develop with a test driven approach
More frequent meetings		

Action items

Rest API MVP sprint

Overview

The past two weeks have consisted of improving the rest service that was started in the last sprint, as well as adding data visualization features as requested by Seaonics.

Date	11 Mar 2022
Team	Wirelogger
Participants	David Rise Knotten einar

Retrospective

Start doing	Stop doing	Keep doing
More frequent standups	Spend too much time at work	Being inovative

Action items

Rigidity Sprint

Overview

The past two weeks have been spent on improving the rigidity of the system and fixing errors to improve the overall feel of the system

Date	28 Mar 2022
Team	Wirelogger
Participants	David Rise Knotten einar

Retrospective

Start doing	Stop doing	Keep doing
Document code	Spend too much time on minor developer-experience improvements	Good teamwork

Action items

Testing sprint

Overview

This weeks focus has been on getting ready and performing user testing, but due to some unforeseen errors when the group visited the office, the tests have been postponed.

Date	08 Apr 2022
Team	Wirelogger
Participants	

Retrospective

Start doing	Stop doing	Keep doing
More proactive scheduling		Physical meetings at Seaonics HQ

Action items

Testing 2 Sprint

Overview

This sprint the group has performed user testing and focused on improving issues discovered during the tests. Major reworks have also been performed to improve the services performance

Date	22 Apr 2022
Team	Wirelogger
Participants	

Retrospective

Start doing	Stop doing	Keep doing
Aiming towards end goal	Planning new features	User Testing
		Documentation
		Trying new approaches

Action items

Rigidity 2 sprint

Overview

This past sprint has focused on improving the code quality and implementing some new functions through the command and metadata api

Date	06 May 2022
Team	Time-Series Cache
Participants	David Rise Knotten einar

Retrospective

Start doing	Stop doing	Keep doing
		Everything done this sprint, amazing progress in all aspects of the project

Action items

Weekly reports

Status report week 3

Time-Series cache status report

Win

Good progress toward the preliminary report

Needs input

Quite a lot of uncertainty towards what the actual project will be

Focus

Gather information on what the projects goals will be

Status report week 4

Time-Series cache status report

Win

Completed preliminary report

Started looking into development

Signed work agreement (Not 3-part)

Needs input

Which system components will we need to make, which already exist?

Get contracts signed

Focus

Start thinking more about the development process

Status report week 5

Time-Series cache status report

Win

MQTT broker set up

TimescaleDB set up

Wireframes complete

Website structure setup

Cache service able to communicate with MQTT and database

Needs input

No specific issues requiring input

Focus

Setting up skeletons for frontend and cache service

Status report week 6

Time-Series cache status report

Win

Cache service able to cache data

Cache service able to act as middle-man between client and vessel

Needs input

Focus

Setup working data handlers for cached signals for website

Status report week 7

Time-Series cache status report

Win

Caching service up and working, but with some issues

Needs input

Web facing REST API needs planning

Website needs development

Focus

REST API

Status report week 8

Time-Series cache status report

Win

Progress has been fast with most technical aspects laying ahead

Development between REST API and website has been good

Data visualization set up

Needs input

Input from Seaonics about additional features

Focus

Write more documentation on development process

More features for current MVP

Status Report Week 9

Time-Series cache status report

Win

Solid progress on documentation

REST service able to stream data through websocket

Added individual y-axis for each dataset/signal in data visualizaion

Needs input

More detailed use-cases

Focus

Continuing the work on documentation

Bandwidth testing of cache service

Notes	This week there's mostly been documentation focus for the backend, with minimal progress on the product itself
Important Links	

Status report week 10

Time-Series cache status report

Win

All services using encrypted communication

Logging of I/O and metadata in progress

Added data fetching for chart zoom and pan

Improved x-axis of chart to timeseries

Needs input

Use-cases for work-flow design

Focus

Bugfixing / rigidity

Status report week 11

Time-Series cache status report

Win

Rigidity of cache service massively improved

Model/database rework has improved working with database a lot

Needs input

Scales needs re-work as per request of Seaonics

Focus

Utilizing Metadata for caching

Status report week 12

Time-Series cache status report

Win

Double fetching now avoided through the use of metadata

Metadata aggregation functionality close to goal

Needs input

Testing / user testing

Focus

Status report week 13

Time-Series cache status report

Win

Documentation for Requirements

Planning user-testing

Needs input

Feature development

Focus

Improve performance

Status report week 14

Time-Series cache status report

Win

Massive improvements in cache responsiveness

Saving to database times reduced

Website performance with data decimation

Needs input

Focus

Make website UX better

Status report week 15

Time-Series cache status report

Win

Good user tests

API improved

Needs input

Maybe some more user testing

Focus

Add more features, solve issue of infinite requests if cache is down

Status report week 16

Time-Series cache status report

Win

Responsivity massively improved

Healthchecking implemented to let other services know of cache status

State maintaining information about current data streams allowing responsive requests

Command API begun

Performed user testing and got a lot of good feedback and insight

Needs input

Focus

Make site more intuitive / better inform users

Status report week 17

Time-Series cache status report

Win

Error handling massively improved, cache practically un-breakable

Names hashed for security, and implemented state to keep track of name-hash-id mappings

Needs input

Focus

Status report week 18

Time-Series cache status report

Win

Ridgidity for Front- and Backend drastically improved

User Manual almost complete

Command line interface complete for Front- and Backend

MQTT now pooled for both services

Healthcheck refactored into sidecar process to be unhindered by main logic

Needs input

Focus

Status report week 19 and 20

During the weeks 19 and 20, all time has gone to writing the bachelor report.

Delegations of responsibility

Since the team only consisted of two members, the group chose to split the project into two parallel projects, the front-end and the back-end

The projects have run mostly independently with frequent integrations along the way. The projects were divided fully to one of the team members. Therefore, each team member has had full responsibility of their own part of the project. Each team member has run their part as an agile project where all the roles and responsibilities has fallen onto the delegated team member. The exception to this is the roles of customer and user, which were both played by the team at Seaonics.

1.5 Backend and operations

The responsibilities falling on the back-end dev was the development of the cache service, which is the core of the system and needs to work with as few issues as possible. In addition to the main cache, the back-end dev also had the responsibility of developing a REST service which would act like a proxy for the cache, and the management of infrastructure and deployment.

1.6 Frontend

The responsibility of the front-end dev was to create a web-based interface which used data fetched from the REST service to make monitoring and troubleshooting easy as well as accessible. An important aspect of the frontend was also to make visualization and control of the data as intuitive as possible, further improving the processes which would rely on the new system.

Timesheets

David

Week 2

Tuesday 16:00-20:15 Writing pre-project plan

Wednesday 20:45-22:30 pre-project plan

Friday 15:30-19:00 pre-project plan

Week: 9:30

Accumulated: 9:30

Week 3

Tuesday 19:45-23:15 Pre-project plan

Wednesday 18:00-00:00 Confluence / researching

Thursday 10:00-13:45 15:00-16:00 Meeting, pre-project plan

Friday 17:05-20:05 pre-project plan

Saturday 19:00-19:30 pre-project plan

Sunday 15:00-16:15 setting up cache service repo

Week: 19:00

Accumulated: 28:30

Week 4

Monday 17:30-19:15 Meeting

Tuesday 16:15-21:45 pre-project plan

Thursday 14:00-15:00 pre-project plan

Friday 11:00-12:00 17:30-18:45 meeting pre-project plan

Week: 10:30

Accumulated: 39:00

Week 5

Monday 20:00-23:00 setting up database/env etc

Tuesday 16:00-21:15 experimenting with dependencies and docker

Wednesday 18:00-19:00 trying singleton database connection

Thursday: 16:30-18:00 21:45-01:45 DB/MQTT, Started working on handlers

Friday 23:00-02:00 CLI tools/scripts, database setup

Saturday 18:30-20:00 Docker and TLS

Week: 19:15

Accumulated: 58:15

Week 6

Tuesday 13:45-16:00 Working on documentation

Wednesday 18:30-21:00 Working on documentation/researching

Thursday 11:00-12:00 19:45-22:30 Started working with protobuf, some semantic improvements

Friday 11:00-14:45 Meeting and minor improvements and battling with protobuf

Saturday 20:00-23:00 protobuf fixing

Week: 15:15

Accumulated: 73:30

Week 7

Monday 20:00-00:00 Testing protobuf and working with a lot of different things

Tuesday 10:00-16:00 20:30-23:15 A lot of different things

Wednesday 20:00-22:30 dummy data improvement and database loading

Thursday 16:00-19:30 get range/last

Friday 11:00-12:00 13:15-18:50 DataBeans, get range/last

Saturday 12:00-18:00 Fixing issues with duplicated keys

Sunday 14:00-19:15 Loads of bugfixing and different progress

Week: 34:05

Accumulated: 107:35

Week 8

Monday 14:45-16:30 19:30-01:15 Loads of bugfixing, REST service work

Tuesday 19:30-00:00 Input sanitation, bugfixing. REST service

Wednesday 13:00-16:00 18:00-22:30 REST service bugfixing

Thursday 12:00-18:00 Working on REST service

Friday 11:00-15:00 Added websocket support on REST

Week: 27:30

Accumulated: 135:05

Week 9

Monday 15:00-20:00 Report writing

Tuesday 15:00-16:00 20:00-23:15 Report writing

Wednesday 19:00-21:00 Report writing

Thursday 12:45-14:45 19:00-21:00 Report writing

Friday 22:00-23:30 Report writing

Week: 16:45

Accumulated: 151:50

Week 10

Monday 16:45-19:00 Bugfixing and Vessel I/O logging

Tuesday 22:45-01:15 Some messing around with deployment

Wednesday 17:30-19:30 21:00-23:55 Updated deployment, code improvements

Thursday 20:00-23:45 TLS on services, overhauls, aggregator work started

Friday 09:00-12:30 Improving code and meeting

Week: 16:55

Accumulated: 168:45

Week 11

Monday 20:30-00:00 General improvements

Tuesday 18:00-19:45 22:15-00:00 Started working on the cache-fill function

Wednesday 17:30-19:00 20:30-21:30 Improving tests and mqtt mocking

Thursday 17:30-19:00 Working on vessel mock

Week: 11:00

Accumulated: 179:45

Week 12

Tuesday 09:00-10:00 internal meeting 11:30-13:30 Documentation 13:30-14:30 create lightweight vessel mock

Wednesday 19:30-23:45 working on aggregator

Thursday 17:00-20:00 Making use of metadata to avoid double-fetching

Friday 22:00-00:00 fixing bugs and working on “bytecost”

Saturday 16:30-18:00 working on “bytecost”

Week: 14:45

Accumulated: 194:30

Week 13

Monday 12:00-13:00 meeting 20:00-22:00 Documentation

Tuesday 16:00-18:00 Diagrams, report writing

Wednesday 17:00-19:30 I/O logging, aggregator, deployment

Thursday 20:00-22:00 writing report

Sunday 14:00-16:00 21:00-22:00 smoothing bugs, improving tests and JSDoc

Week: 12:30

Accumulated: 207:00

Week 14

Monday 20:00-21:30 documentation

Tuesday 21:40-23:30 report

Wednesday 16:45-19:00 22:45-00:00 preparing test environment, report writing

Thursday 09:00-14:15 workshop at Seaonics

Friday 11:00-15:30 meeting and playing around with database

Saturday 21:00-00:30 database remodeling

Sunday 11:30-14:00 database remodeling, fixing tests

Week: 22:35

Accumulated: 229:35

Week 15

Monday 14:30-16:00 22:30-01:00 documentation/report

Tuesday 19:00-22:00 Refactoring and documentation

Thursday 12:00-15:00 19:00-21:30 Documentation and testing, Improving insertions

Friday 13:00-18:00 documentation, testing

Saturday 06:00-10:15, 22:00-01:50 Data streaming to database, testing, improving speeds

Sunday 16:00-19:00 22:30-01:15 refactoring, improving performance, local check on currently streaming intervals

Week: 31:20

Accumulated: 260:55

Week 16

Monday 17:00-21:00 Updating protobuf definition, improving streaming rigidity, system documentation

Tuesday 20:30-23:15 Testinng, working on presentation

Thursday 12:00-14:20 21:30-02:05 user testing, Healthcheck implementation, complete API overhaul, start of command API

Friday 11:00-16:00 23:00-02:15 Meeting, Improving responsivity, readmes, REST routes, Command API on REST

Saturday 15:00-21:25 22:40-23:40 23:50-00:45 deploying on new machines, hashing names, centralized name/id/hash mapping, documentation

Sunday 13:00-22:05 fixing bugs, cache seems failproof, documentation, fixing tests

Week: 39:20

Accumulated: 300:15

Week 17

Monday 14:30-17:00 18:30-23:00 documentation, improving reliability on big requests, testing

Tuesday 14:30-16:00 21:00-00:05 documentation, dockerizing, StreamStateController

Wednesday 17:30-19:30 22:30-01:40 Centralizing logic in streamstate and entitydata controllers

Thursday 17:35-19:40 22:15-01:35 improving error propagation

Friday 20:10-23:45 Documentation, edge case bugfixing, testing prisma DB client

Saturday 17:30-19:00 21:00-22:55 Report writing, tests

Sunday 14:00-21:35 Documentation, linting, docker compose

Week: 36:45

Accumulated: 337:00

Week 18

Monday 17:00-20:00 Documentation/report

Tuesday 14:00-22:00 Documentation, MQTT pooling, diagrams

Wednesday 17:00-20:45 Documentation, Healthcheck refactored to sidecar process

Thursday 20:30-01:15 MQTT pooling on cache, fixing pg-pool ghost errors

Friday 10:30-13:15 20:50-01:05 Deployment and meeting, fully implementing wipecache, fixing tests after MQTTPool, improving test coverage

Saturday 19:00-21:00 writing documentation

Sunday 13:00-15:30 documentation

Week: 31:00

Accumulated: 368:00

Week 19

Monday 17:30-20:30 report writing, improving test-data generation

Tuesday 22:00-01:00 report

Wednesday 20:00-23:30 report

Friday 19:00-21:35 report

Saturday 15:00-18:00 report

Sunday 12:55-16:10 18:50-21:10 report, bugfixing and final deployment

Week: 20:40

Accumulated: 388:40

Week 20

Monday 18:45-23:50 report

Tuesday 11:30-16:00 20:00-23:30 report

Wednesday 15:35-23:45 report

Week: 21:05

Accumulated: 409:45

Einar

Week 2

Tuesday 16:00-20:00 writing pre-project plan

Wednesday 20:00-22:00 pre-project plan

Friday 15:00-19:00 pre-project plan

Week: 10

Accumulated: 10

Week 3

Thursday 10:00–14:00 15:00–16:00 meeting, pre-project plan

Friday 12:00–20:00 setting up Next.js project + starting building

Saturday 12:00–16:00 continue building + tech research

Sunday 12:00–16:00 implemented pages and sidebar to website

Week: 21

Accumulated: 31

Week 4

Monday 17:30-19:15 meeting

Tuesday 16:00-22:00 pre-project plan

Thursday 14:00-15:00 pre-project plan

Friday 11:00-12:00 17:30-18:45 meeting pre-project plan

Week: 11

Accumulated: 42

Week 5

Monday 10:00-13:00 research

Tuesday 12:00-16:00 documentation

Week: 7

Accumulated: 49

Week 6

Tuesday 10:00-16:00 documentation

Friday 11:00-12:00 meeting

Week: 5

Accumulated: 54

Week 7

Monday 10:00-16:00 documentation

Wednesday 10:00-16:00 documentation

Week: 12

Accumulated: 66

Week 8

Tuesday 20:00-22:00 implemented API calls to REST service

Wednesday 12:00-21:00 implemented vessel / signal router + Chart.js with Pan & Zoom

Thursday 14:00-18:00 implemented calling for last N datapoints via REST service and adding to chart

Friday 20:00-22:00 implemented calling for specific range via REST service + multiple axis

Saturday: 13:00-17:00 started Cypress testing + manual chart control buttons

Week: 21

Accumulated: 87

Week 9

Monday 13:00-16:00 testing different axis options for Chart.js

Thursday 11:00-15:00 implemented stepped lines for datasets

Friday 10:00-17:00 implemented seamless data fetching when panning in x-direction

Week: 14

Accumulated: 101

Week 10

Monday 19:00-21:00 refactored support for multiple signals when interacting with chart

Thursday 10:00-17:00 documentation

Week: 9

Accumulated: 110

Week 11

Monday 17:00-21:00 refactoring

Thursday 18:00-22:00 refactoring and added sorting of timestamps after fetch

Week: 8

Accumulated: 118

Week 12

Tuesday 12:00-16:00 implementing animated counter for byte- cost and transfer

Thursday 10:00-15:00 refactoring + report

Week: 9

Accumulated: 127

Week 13

Monday 12:00-13:00 meeting

Friday 10:00-15:00 refactoring

Sunday 17:00-21:00 improvement of chart component

Week: 10

Accumulated: 137

Week 14

Monday 10:00-17:00 refactoring prop drilling into context
Wednesday 10:00-16:00 query and param management re-work
Thursday 09:00-14:00 workshop at Seaonics

Week: 18
Accumulated: 155

Week 15

Monday 10:00-15:00 documentation/report
Tuesday 10:00-16:00 refactoring prop drilling into context

Week: 11
Accumulated: 166

Week 16

Tuesday 09:00-1800 working on individual settings for datasets + API re-work
Wednesday 12:00-15:00 working on toast / notification system
Thursday 12:00-14:00 user testing with Seaonics
Friday: 10:00-18:00 rigidity fixes for website + date picker added
Saturday: 18:00-21:00 reworked down sampling to manual version

Week: 30
Accumulated: 196

Week 17

Tuesday 09:00-1800 working on JSDocs for website + documentation/report
Wednesday 14:00-18:00 first version of Command Line page
Thursday 17:00-19:00 metadata fetch command line working

Week: 15
Accumulated: 211

Week 18

Monday 16:00-19:00 refactor with ESLint website
Tuesday 10:00-14:00 documentation/report
Thursday 20:00-22:00 JSDocs completed
Friday 13:00-17:00 refactoring from supervisor comments
Saturday 12:00-18:00 added debounce / rate-limit to buttons for chart panning

Week: 19
Accumulated: 230

Week 19

Monday 12:00-16:00 documentation/report

Tuesday 12:00-20:00 documentation/report + refactoring

Thursday 12:00-16:00 documentation/report

Friday 12:00-16:00 documentation/report

Saturday 12:00-16:00

Week: 24

Accumulated: 254

Week 20

Monday 10:00-16:00 documentation/report

Tuesday 10:00-13:00 documentation/report + refactoring

Wednesday 10:00-16:00 documentation/report

Thursday 10:00-20:00 documentation/report

Week: 25

Accumulated: 279