

# Standort-basierte Daten via MQTT

Seminararbeit “Webservice Security”

Marco Wettstein

2015-05-14

## Contents

<b>Einleitung</b>	<b>2</b>
<b>Ausgangslage</b>	<b>3</b>
Timetraces . . . . .	3
Standortverlauf als weitere Event-Quelle . . . . .	3
OwnTracks . . . . .	4
<b>Zielsetzung</b>	<b>7</b>
<b>Recherche</b>	<b>8</b>
Location-Apps & -Dienste . . . . .	8
Owntracks . . . . .	8
Google+ . . . . .	8
. . . . .	8
MQTT . . . . .	8
Broker . . . . .	8
Topics und Subscriptions . . . . .	8
Wahl eines Brokers . . . . .	8
Setup . . . . .	8
<b>Design</b>	<b>9</b>
Sicherheitsaspekte . . . . .	9
Sensible Daten . . . . .	9
Verschlüsselung . . . . .	9
Authentifizierung . . . . .	9
Architektur . . . . .	9

<b>Umsetzung</b>	<b>10</b>
Screenshots . . . . .	10
<b>Diskussion</b>	<b>11</b>
<b>Ausblick</b>	<b>12</b>
Indoor-Standorte mittels Beacons . . . . .	12
Generischer Event-Service . . . . .	12
 <b>Einleitung</b>	

# Ausgangslage

## Timetraces

Im Rahmen einer Seminararbeit wurde für die Controlling- und Zeiterfassungs-Applikation “controllr” (siehe Abbildung ??) eine neue Client-Anwendung gebaut, welche durch die Integration verschiedener Dienste wie Github, Redmine und Google Calendar eine Art Protokoll der geleisteten Arbeit erstellt. Aus den Einträgen dieses Protokoll können in der Anwendung direkt Zeiteinträge in “controllr” erstellt werden. Abbildungen ?? zeigt das Arbeitsprotokoll von “timetraces”. Durch Anwählen eines Eintrages wird eine vor-ausgefüllte Eingabemaske angezeigt, welche den Zeiteintrag über eine REST-Schnittstelle an “controllr” sendet (Abbildung 3).

### Daily entries

February 2015

Mo	Tu	We	Th	Fr	Sa	Su
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	

Project

abc-123 - Project ABC

Task

Development

Start

16:35

End

17:45

Duration

☒ Billable

Create Entry

Description

Implement new Feature X

### Entries for 10 Feb 2015

Project	Project desc.	Task	Description	Start	End	Duration	Billable	Edit	Copy	Delete
abc-123	Projekt ABC	Internal Meeting	Kickoff Meeting	09:15	10:45	1.50	✓	✎	📋	🗑
xyz-001	Projekt XYZ	Development	Refactor I18n-Module	11:45	12:15	0.50	✓	✎	📋	🗑
xyz-002	Project XYZ Support	Development	support user	13:05	13:20	0.25	✓	✎	📋	🗑
abc-123	Project ABC	Development	Implement that feature, solve a...	13:20	16:20	3.00	✓	✎	📋	🗑
xyz-001	Project XYZ	Internal Meeting	Code Review	16:25	16:35	0.17	✓	✎	📋	🗑
Incurred						5.42				

Showing 1 to 5 of 5 entries

Figure 1: Screenshot von “Controllr” (Quelle [sem\_handheld, p. 6])

“Timetraces” wurde als “Meteor”-Anwendung gebaut (siehe dazu Abschnitt ??) und ist eine Client-Server-Anwendung, welche externe Dienste integriert. Die Anwendung speichert dabei ausser den Benutzer-Logins und den Einstellungen der Benutzer keine weiteren Daten. Sämtliche Daten werden dabei vom Server der Anwendung aggregiert und an den Client gesendet. Die Daten werden vom Server dabei über REST-Schnittstellen in einem Polling-Verfahren abgerufen. Das Polling wird gestartet, sobald der clientseitige Teil der Anwendung die Daten über eine DDP-Subscription abonniert und beendet, sobald der Client die Subscription beendet.

Abbildung 4 zeigt den Ablauf einer Subscription eines Clients.

## Standortverlauf als weitere Event-Quelle

“TimeTraces” nutzt bisher *Github*, *Google Calendar* und *Redmine* als Event-Quellen. Als weitere Event-Quelle soll nun der Standort-Verlauf des Benutzers genutzt werden. Diese Daten sollen dem Benutzer helfen, die Zeiteinträge genauer zu erfassen. Der Benutzer sieht somit nicht nur, was wann gearbeitet wurde, sondern auch wo. Es löst zudem das Problem, dass es oft schwierig ist, sich an den

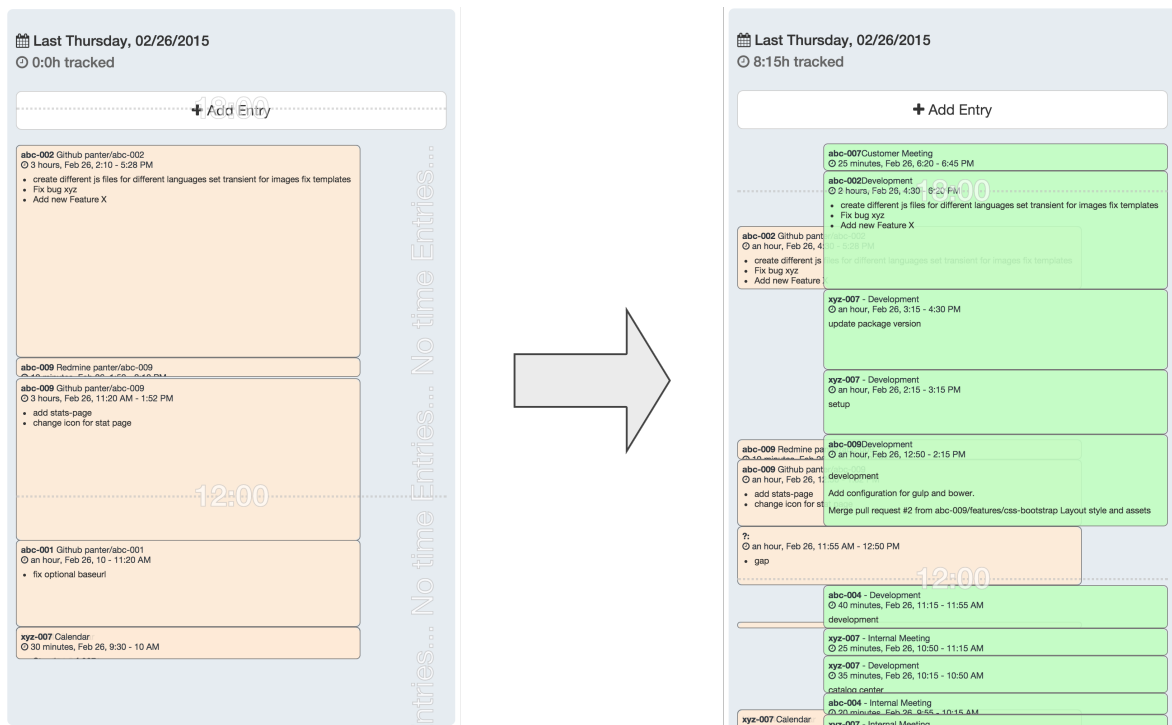


Figure 2: Darstellung der Event-Liste eines Tages in “timeTraces” (Quelle [sem\_handheld, p. 22])

Startzeitpunkt einer Arbeit zu erfassen: es kann z.b. festgestellt werden, wann das Büro betreten wurde.

## OwnTracks

Owntracks wurde als Ersatz für den eingestellten Google Standort-Dienst “Latitude” entwickelt und ursprünglich in Anlehnung an das Vorbild und dem Verwendeten Protokoll als *MQTTitude* bezeichnet.<sup>1</sup>

Die Anwendung zeichnet den Standort des Benutzers im Hintergrund auf und sendet die Daten an einen zu definierenden MQTT-Broker unter einem wählbaren *Topic* (Siehe Abschnitt ). Dabei können verschiedene Einstellungen wie die Häufigkeit der Standortprotokollierung

OwnTracks ist als quelloffene Anwendung für IOS und Android erhältlich und ist unter der *Eclipse Public Licence* veröffentlicht.<sup>2</sup>

Die Verwendung von *OwnTracks* ist im Rahmen dieser Arbeit als Ausgangslage vordefiniert und gibt das Protokoll *MQTT* vor, es sollen aber auch Alternativen betrachtet werden<sup>3</sup>

<sup>1</sup>Siehe Quelle [git\_owntracks]

<sup>2</sup>Lizenz und Quelle unter [licence\_owntracks]

<sup>3</sup>Owntracks nutzt MQTT als Übertragungsprotokoll. Es ist aber auch denkbar, dass anderen Anwendungen ein anderes Protokoll verwenden.

new Time Entry

Project ID

abc-002 Backen...

Task ID

Development

☒ Billable

Description

- create different js files for different languages set transient for images fix templates  
- Fix bug xyz  
- Add new Feature X

Date

26.02.2015

Start

14:10

End

17:28

User ID

84

Close

Insert

Figure 3: Eingabemaske für einen Zeiteintrag in “timeTraces”. Alle Felder werden vorausgefüllt (Quelle [sem\_handheld, p. 23])

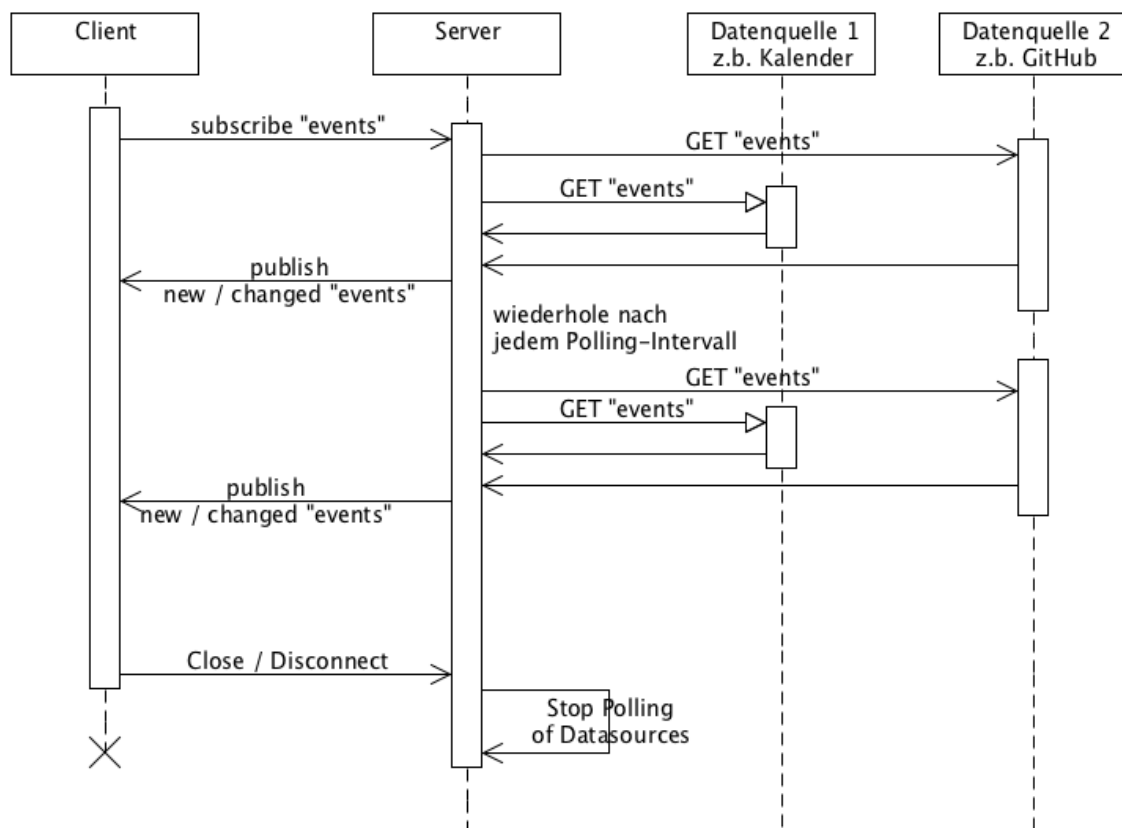


Figure 4: Ablauf einer Subscription von “timeTraces” zwischen Client - Server und externen Quellen (Quelle [sem\_handheld, p. 19])

# Zielsetzung

Die Ziele dieser Arbeit sind

- Vertiefung in das Thema MQTT
- Betrachtung sicherheitsrelevanter Aspekte von MQTT,
- Betrachtung generell sensibler User-Daten, wie Standortverlauf
- Setup einer geeigneten MQTT-Broker-Lösung mit geeigneten Sicherheitseinstellungen
- Verbinden von “OwnTracks” oder einer ähnlichen Anwendung und “TimeTraces” via MQTT und dem gewählten Broker

## Recherche

### Location-Apps & -Dienste

Owntracks

tODO: runter schieben?

Google+

## MQTT

Broker

Topics und Subscriptions

Wahl eines Brokers

Setup



# Design

## Sicherheitsapakte

Sensible Daten

Verschlüsselung

Authentifizierung

## Architektur

DDP, Datenspeicherung auf timetraces

**Umsetzung**

**Screenshots**

## Diskussion

**Ausblick**

**Indoor-Standorte mittels Beacons**

**Generischer Event-Service**