

Bachelorarbeit, Abteilung Informatik

# **Mobile Performance**

## **Implementation neuer statistischer Auswertungsmöglichkeiten und Überprüfung der Aussagekraft von Netztest-Kampagnen**

Frühjahrssemester 2017

Autoren: Andrea Hauser und David Windler  
Betreuer: Prof. Dr. Peter Heinzmann

Arbeitsperiode: 20.02.2017 – 16.06.2017  
Arbeitsumfang: 360 Arbeitsstunden bzw. 12 ECTS pro Student  
Link: <http://sinv-56041.edu.hsr.ch/api>

## Abstract

Mobile Provider, wie Swisscom, Sunrise und Salt, lassen sich regelmässig von Testorganisationen bewerten, um mit den Ergebnisse zu werben. Ziel dieser Bachelorarbeit war es, die Einflüsse verschiedener Parameter auf die Rangierungen aufzuzeigen und mittels automatischer Klassierungen der Speedtest-Messungen aussagekräftigere Ranglisten zu ermöglichen

Anhand der Analyse bekannter Netztests (connect, Ookla und weitere Anbieter) wurden deren Vor- und Nachteile aufgezeigt. Für Speedtest-Apps wurden Möglichkeiten zur automatischen Klassierung von Messorten und Bewegungsart untersucht. Es entstand ein Web API, welches mittels Lokationsangaben von mindestens drei zusammengehörenden Messpunkten eine Klassierung von Messungen ermöglicht. Abschliessend wurde aufgezeigt, wie die Klassierung zu aussagekräftigeren Ranglisten führt bzw. unter welchen Bedingungen die Rangierungen bei Netztest-Kampagnen nicht mehr zufällig sind.

Der entwickelte Algorithmus leitet aus den bestehenden Messdaten der cnlab AG, dem freien Kartenmaterial von OpenStreetMap sowie aus Geographie- und Raumdaten des Bundes die folgenden Klassierungen ab:

- Messort (Schiene, Strasse, Gebäude)
- Fortbewegungsart (stationär bis sehr schnell fahrend)
- Bevölkerungsdichte und Gemeindetyp
- Geografische Umgebung (Wald, Wiese, Wohngegend)

Die Klassierung basiert auf einer Kombination der Abstände von Messorten zu Objekten, der Genauigkeit der Lokationsangaben und der Fortbewegungsgeschwindigkeit während einer Messung. Mit dem Algorithmus konnten 82% der Messungen mit einer Ortsauflösung von besser als 200 Meter korrekt klassiert werden. Die statistischen Auswertungen ergaben, dass bei Gegenüberstellung des 50ten Perzentils aller Messungen eines Providers zu den 50ten Perzentilen der einzelnen Monate Abweichungen von bis zu -40,53% auftreten.

# Aufgabenstellung



## **Bachelorarbeit «Mobile Performance - Implementation neuer statistischer Auswertungsmöglichkeiten und Überprüfung der Aussagekraft von Netztest-Kampagnen»**

---

Studiengang: Informatik (I)

---

Institut: Institut für Networked Solutions  
Gruppe: Andrea Hauser und David Windler

---

Betreuer: Prof. Dr. P. Heinzmann  
Korreferent: Prof. St. Keller, HSR  
Experte: Dr. Th. Siegenthaler, CSI  
Industriepartner: -

## **Ausgangslage**

---

Mobile Provider lassen sich jährlich von Testorganisationen bewerten, um die Ergebnisse marketingtechnisch auswerten zu können. Am bekanntesten ist der jeweils im Herbst durchgeführte Test der deutschen Zeitschrift Connect, bei welchem vor allem Testfahrten auf im Voraus bekannt gegebenen Strecken durchgeführt werden. Andere Tests basieren auf dem Crowd-Sourced-Ansatz (Oakla, opensignal). Das Bilanz Rating bewertet eher administrative Aspekte.

Welche Aussagekraft diese empirischen Untersuchungen tatsächlich besitzen und welche Schlussfolgerungen daraus gezogen werden können (z.B. die Erstellung von Ranglisten), ist fraglich. Einerseits sind die Unterschiede zwischen den Anbietern mittlerweile sehr gering. Andererseits können je nach geografischer Region, Durchführungsort der Tests (z.B. in einem Gebäude, im Auto, im Zug, im Freien) oder Bewegungsart (stationär, langsam oder schnell bewegt) völlig unterschiedliche Ergebnisse resultieren.

## Ziel

---

Es soll ein besseres Verständnis zur Aussagekraft der verschiedenen Speedtests erarbeitet werden. Für die cnlab Mobile Performance Benchmarking (MPB) Anwendung sind erweiterte Auswertemöglichkeiten der Ergebnisse (Erkennung von Messort bzw. Messumgebung und Fortbewegungsart) zu realisieren. Mittelfristig soll anhand dieser Erweiterungen die statistische Aussagekraft der Tests quantitativ beschrieben und besser interpretiert werden können.

## Aufgaben

---

Unter anderem stehen folgende Teilaufgaben zur Diskussion.

1. Im Rahmen der **Einarbeitung**/Analyse lernen Sie die cnlab Speedtest App (Android Version) und die aktuellen Auswertemöglichkeiten von Crowd Sourced- Messansätzen genau kennen. Dabei vergleichen Sie verschiedene Mobile Benchmarking Messansätze in Bezug auf Aspekte wie:
  - a. Funktionalität (insbesondere UX-Funktionen)
  - b. Android App Architektur, erfasste Parameter, Messprinzipien
  - c. Auswertemöglichkeiten, Auswerteplattform (Open Data Approach)
  - d. Statistische Angaben
  - e. Administrative Angaben (Firma, Nutzer, Finanzierung)
  - f. Erfahrungen aus eigenen Tests, Beurteilung
2. **Automatische Klassierung von Messungen**: Entwicklung von Auswerte-Anwendungen zur genaueren Charakterisierung der Speedtest App Ergebnisse. Anhand der erfassten GPS- und Bewegungsdaten soll die Anwendung erkennen
  - a. ob eine Messung in einem Gebäude, auf der Schiene (Zug, Tram) oder auf der Strasse (Auto/Bus/Postauto) durchgeführt wurde und
  - b. ob der Test stationär oder in Bewegung durchgeführt wurde.
  - c. Bestimmung von Mapping Kriterien (z.B. Geschwindigkeit, Signalumgebung, Beschleunigungen/Erschütterungen, WLAN SSIDs)
  - d. Nutzung von Systemen wie OpenStreetMap, OpenRailwayMap Layer und OpenLayers.
3. Entwicklung von **statistischen Tests und Bewertungskriterien zur Aussagekraft von Mobile Benchmarking Untersuchungen**
  - a. Quellen, Gründe für Falschangaben
  - b. Unter welchen Bedingungen sind Ergebnisse nicht mehr «zufällig»
  - c. Methoden zur Auswertung
  - d. Quantitative Beschreibung der Aussagekraft

## Referenzen

---

1. Hinweise zur Durchführung von Studienarbeiten  
[https://www.dropbox.com/s/v60mbzt393ho23g/SABA-Web-Anleitungen\\_Heinzmann.zip?dl=0](https://www.dropbox.com/s/v60mbzt393ho23g/SABA-Web-Anleitungen_Heinzmann.zip?dl=0)
2. BAKOM <https://www.admin.ch/gov/de/start/dokumentation/medienmitteilungen.msg-id-64764.html>
3. Broadband Europe <https://ec.europa.eu/digital-single-market/en/broadband-europe>
4. Netztests von Connect <http://www.connect.de/specials/netztest/>  
<http://www.connect.de/vergleich/netztest-2017-bestes-handy-netz-schweiz-mobilfunk-3196571.html>
5. Speedtest.net Awards <http://www.speedtest.net/awards/ch/carrier>
6. Bilanz Telekomanbieter Rating <http://www.bilanz.ch/unternehmen/das-sind-die-besten-telekomanbieter-2016-718100>
7. OpenStreetMap <http://wiki.openstreetmap.org/wiki/Railways>
8. OpenRailwayMap <http://wiki.openstreetmap.org/wiki/OpenRailwayMap>
9. OpenLayers <https://openlayers.org/>
10. Geodaten <http://geops.de/>
11. Geodaten SBB  
<http://tracker.geops.ch/?z=12&s=1&x=974857.5905&y=5982760.1694&l=transport>

Rapperswil,  
Ort, Datum

---

Betreuer

---

## Management Summary

Mobile Provider, wie Swisscom oder Sunrise, lassen sich jährlich von bekannten Netztestanbietern bewerten. Anschliessend werben sie stark mit dem Ergebnis, da alle Provider entweder mit „sehr gut“ oder sogar „Testsieger“ ausgezeichnet werden. Es ist zu untersuchen, wie stark man sich auf die Aussage von solchen Netztests verlassen kann. Weiter soll herausgefunden werden, unter welchen Bedingungen Speedtest-Ergebnisse statistisch aussagekräftig sind.

Für die Durchführung der Netztests werden teilweise auch Speedtest-Apps verwendet. Mithilfe von diesen Apps lässt sich die Geschwindigkeit des mobilen Internets bestimmen. Diese hängt jedoch davon ab, unter welchen Bedingungen die Messung durchgeführt wird. Je nachdem ob die Messperson mit einer hohen Geschwindigkeit im Zug unterwegs ist oder bei sich zu Hause sitzt, resultieren andere Ergebnisse. Um herauszufinden, wo sich die Messperson befindet, wird eine automatische Klassierung benötigt.

Zu Beginn wurden Ergebnisse von Netztests untersucht, um Stärken und Schwächen jedes einzelnen aufzudecken. Zudem wurden bekannte Speedtest-Apps untersucht.

Es folgten die Abklärungen nach bestehenden Programmen für das Problem der Messdatenklassierung. Da keine befriedigende Lösung gefunden werden konnte, wurde selbst ein entsprechender Algorithmus erarbeitet und im Code umgesetzt. Der Algorithmus wurde im Verlauf des Projekts Schritt für Schritt verbessert.

Gegen den Schluss des Projekts wurden die Speedtest-Messungen statistisch untersucht. Dazu wurden Messdaten von neun Monaten mit Excel-Tabellen ausgewertet. Es wurde für jeden Provider geprüft, ab wann zufällig ausgewählte Daten als zuverlässig einzustufen sind. Zudem wurden Unterschiede zwischen den einzelnen Monaten sowie zwischen Hauptstadt- und Kanton analysiert.

Durch die Untersuchung der Netztests wurde festgestellt, dass deren Messmethoden sehr unterschiedlich sind. Aus diesem Grund können die Tests untereinander nur schwer verglichen werden. Zudem ist dies der Grund dafür, dass es für jeden Provider einen Netztest gibt, mit dem er anschliessend Werbung schalten kann.

Das Ergebnis der statistischen Auswertung ist, dass zwischen den einzelnen Monaten starke Schwankungen vorliegen. Auch die Unterschiede zwischen städtischen und ländlichen Regionen sind gross. Darum wird empfohlen, dass bei solchen Untersuchungen immer Messungen des gesamten Zeitraums sowie verschiedener Regionen enthalten sind. Zudem wurden die Anzahl Werte bzw. der Anteil pro Provider bei einer zufälligen Auswahl erarbeitet, welche zu einer zuverlässigen Aussage führen.

Mit dem Tagging-Server wurde eine automatische Kategorisierung von Speedtest-Messungen umgesetzt. Mit seiner Hilfe können differenziertere Auswertungen erstellt werden.

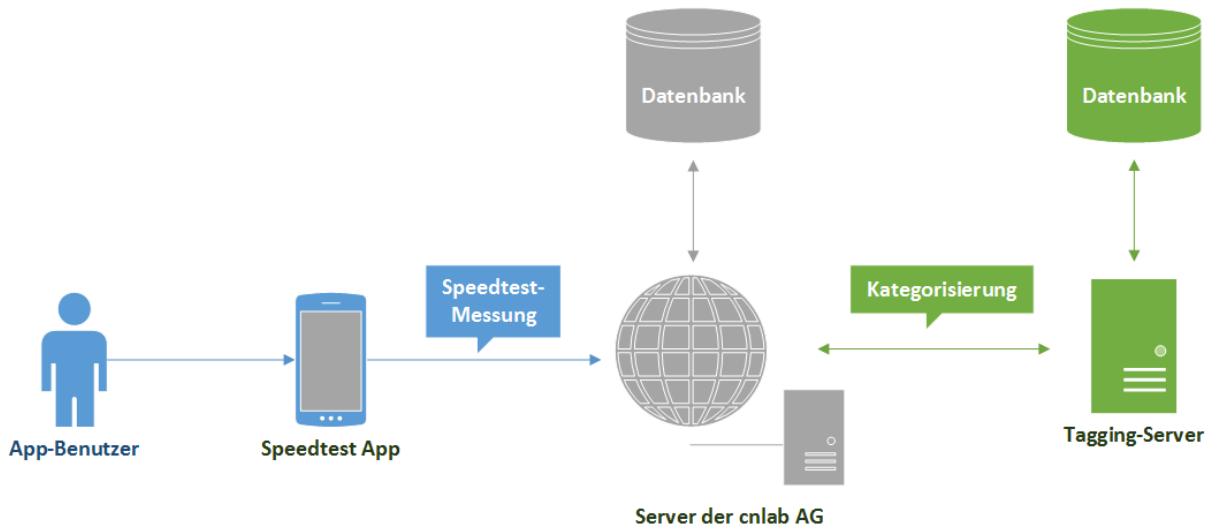


Abbildung 1: Ablauf einer Speedtest-Messung mit automatischer Kategorisierung

In Abbildung 1 wird der Ablauf einer solchen Speedtest-Messung mit Kategorisierung aufgezeigt. Der App-Benutzer startet eine Messung, deren Resultat anschliessend an den Hersteller der Speedtest-App, z.B. die cnlab AG, weitergeleitet wird. Diese schickt anschliessend einen Teil der Messdaten an den Tagging-Server. Der Server führt die Kategorisierung durch und sendet diese an die cnlab AG zurück.

Der Tagging-Server kategorisiert ca. 80% der angenommenen Daten korrekt. Es gilt nun, die restlichen Fälle genauer zu analysieren und Lösungsansätze auszuarbeiten.

# Inhaltsverzeichnis

<b>I</b>	<b>Hauptbericht</b>	<b>10</b>
<b>1</b>	<b>Einleitung</b>	<b>11</b>
1.1	Grundbegriffe . . . . .	13
<b>2</b>	<b>Analyse</b>	<b>14</b>
2.1	Aussagekraft von Netztests . . . . .	14
2.2	Auswertung zu Speedtest-Apps . . . . .	19
2.3	Auswertung zur cnlab Speed Test-App . . . . .	30
<b>3</b>	<b>Statistische Auswertungen</b>	<b>32</b>
3.1	Einführung in die Statistik . . . . .	32
3.2	Zufallsexperiment . . . . .	38
3.3	Monatliche Auswertung . . . . .	44
3.4	Regionale Auswertung . . . . .	46
3.5	Schlussfolgerungen . . . . .	46
<b>4</b>	<b>Verwendete Datenquellen</b>	<b>48</b>
4.1	Lizenzabklärungen OpenStreetMap (OSM) . . . . .	49
4.2	Lizenzabklärung GeoAdmin API . . . . .	49
<b>5</b>	<b>Software Engineering</b>	<b>51</b>
5.1	Routes (Layer 1) . . . . .	53
5.2	Business_logic (Layer 2) . . . . .	53
5.3	Persistence (Layer 3) . . . . .	55
5.4	Config (Layer 4) . . . . .	56
5.5	npm-Module . . . . .	56
<b>6</b>	<b>Realisierung</b>	<b>57</b>
6.1	Use Case: Mapping Geodaten auf Bewegungseigenschaften . . . . .	58
6.2	Input-Daten . . . . .	61
6.3	Output-Daten . . . . .	70
6.4	Logik . . . . .	73

6.5	Performance-Optimierungen . . . . .	86
<b>7</b>	<b>Weiterführende Arbeiten</b>	<b>88</b>
7.1	Arbeiten am Tagging-Server . . . . .	88
7.2	Arbeiten an der cnlab App . . . . .	90
7.3	Arbeiten an der Auswertung zu Speedtest-Apps . . . . .	91
<b>8</b>	<b>Schlussfolgerung</b>	<b>92</b>
<b>9</b>	<b>Literaturverzeichnis</b>	<b>93</b>
<b>10</b>	<b>Abkürzungsverzeichnis (Glossar)</b>	<b>102</b>
<b>II</b>	<b>Anhang</b>	<b>104</b>
<b>11</b>	<b>Ablageorte</b>	<b>105</b>
11.1	Arbeitsdokumente . . . . .	105
11.2	Code . . . . .	106
<b>12</b>	<b>Projektmanagementplan</b>	<b>107</b>
12.1	Kostenvoranschlag . . . . .	107
12.2	Phasen / Iterationen . . . . .	107
12.3	Meilensteine . . . . .	111
12.4	Zeiterfassung . . . . .	111
<b>13</b>	<b>Risikomanagement</b>	<b>114</b>
13.1	Umgang mit Risiken . . . . .	114
<b>14</b>	<b>Details zur Lösungsfindung</b>	<b>115</b>
14.1	Installation Tagging-Server . . . . .	115
14.2	Verwendete Libraries / npm-Module . . . . .	116
14.3	Auswertung der location-Korrektheit . . . . .	117
14.4	Last-Tests . . . . .	126
14.5	Unit-Tests . . . . .	128
14.6	Produktabklärungen . . . . .	128
14.7	Prototyp und Versionen . . . . .	128
14.8	Lizenzabklärungen . . . . .	129
<b>15</b>	<b>Plakat zur Bachelorarbeit</b>	<b>136</b>
<b>16</b>	<b>Verwendete Werkzeuge</b>	<b>138</b>
16.1	Dokumentenverwaltung . . . . .	138
16.2	Server-Zugriff . . . . .	139

16.3	Projektverwaltung . . . . .	139
16.4	Dokumentation . . . . .	140
16.5	Software-Entwicklung . . . . .	141

**Teil I**  
**Hauptbericht**

# Kapitel 1

## Einleitung

Mobile Provider, wie z.B. Swisscom, werben stark mit Auszeichnungen, welche ihnen jährlich von bekannten Netztest-Anbietern verliehen werden. Dabei fällt auf, dass alle Provider „sehr gut“ oder sogar „Testsieger“ sind. Wie vertrauenswürdig sind also die oben beschriebenen Auszeichnungen und wie ist es möglich, dass pro Test unterschiedliche Resultate hervorkommen? Lohnt sich ein Wechsel zum Testsieger?

Um diese Fragen zu beantworten, werden im Rahmen dieser Arbeit solche Netztests analysiert.

Ab wann sind solche Untersuchungen zu Mobile Speedtest-Messungen aussagekräftig? Haben die im Netztest festgelegten Fahrtrouten einen Einfluss auf das Resultat? Wie zuverlässig können Auswertungen zu einzelnen Monaten gemacht werden?

Diese statistischen Fragen werden in dieser Arbeit mit Hilfe von Mobile Speedtest-Daten der cnlab AG untersucht.

Um die Resultate von Speedtest-Messungen besser einordnen zu können, ist es notwendig, diese nach bestimmten Kriterien zu kategorisieren. Fand eine Messung in der Stadt oder auf dem Land statt? War die Testperson in Bewegung? Fand die Messung auf der Strasse oder in einem Gebäude statt?

Im Rahmen dieser Arbeit wurde ein Algorithmus entwickelt, welcher diese Kategorisierung automatisch durchführt. Auf der folgenden Abbildung ist eine Übersicht über den erstellten Server aufgeführt, welcher diesen Algorithmus implementiert. Der erstellte Server ist grün markiert und wird nachfolgend Tagging-Server genannt.

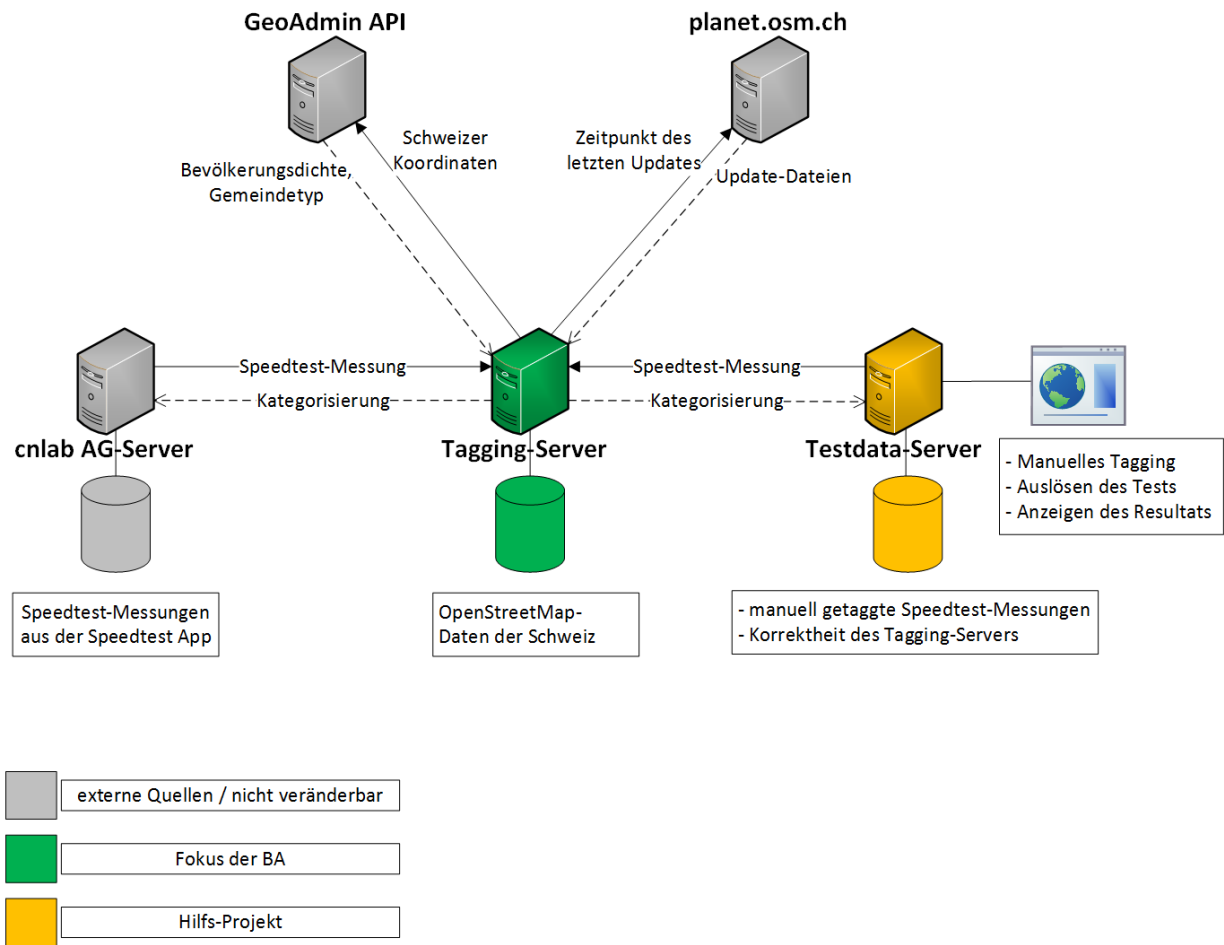


Abbildung 1.1: Übersicht der erstellten Anwendungen

Der Tagging-Server erhält Speedtest-Messung von der cnlab AG (grau), welche ihre Messungen kategorisieren möchte. Der Server holt sich seine benötigten Daten aus der lokalen Datenbank, welche die aktuellen OpenStreetMap-Daten der Schweiz beinhaltet. Diese Daten werden stündlich über planet.osm.ch aktualisiert. Weitere Informationen holt sich der Tagging-Server vom GeoAdmin API des Bundes [1], um schlussendlich mit allen Daten eine Kategorisierung vorzunehmen und diese an die cnlab AG zurückzusenden.

Um die Genauigkeit des Servers zu überprüfen, wurde zusätzlich der Testdata-Server (gelb) implementiert, welcher von Hand kategorisierte Daten mit den Antworten des Tagging-Servers vergleicht.

In folgendem Abschnitt werden die wichtigsten Begriffe erklärt, welche in dieser Dokumentation vorkommen.

## 1.1 Grundbegriffe

- **Mobile Provider:** Organisation, welche ihren Kunden eine kabellose Übertragung von Telefongesprächen und Internet-Daten zur Verfügung stellt.
- **Netztest:** Test von Mobilnetzen verschiedener Mobile Provider nach bestimmten Kriterien. Diese Kriterien sind je nach Netztest-Anbieter unterschiedlich definiert.
- **OpenStreetMap (OSM):** Projekt mit dem Ziel, eine frei zugängliche Weltkarte zu erschaffen. [2]
- **Referenz-Server:** Gegenstelle zum Testgerät bei der Durchführung eines Speedtests. Wird z.B. ein Download ausgeführt, so stammen die Daten von diesem Server.
- **Speedtest:** Test, welcher unter anderem die Upload- und Download-Datenrate eines Gerätes misst.
- **Speedtest-Messung:** Resultat der Durchführung eines Speedtests.
- **Tagging:** Kategorisierung von Speedtest-Messungen nach gewissen Kriterien.
- **Tagging-Server:** In dieser Arbeit erstellter Webserver zur automatischen Kategorisierung nach den Kriterien Messort, Fortbewegungsart, Bevölkerungsdichte, Gemeindetyp und geografische Umgebung.
- **Upload- und Download-Datenrate:** Datenmenge, welche in einer bestimmten Zeit übertragen werden kann. Beim Upload werden diese Daten an einen Empfänger übertragen, beim Download erhält man diese Daten von einem Sender.

Im folgenden Kapitel werden bekannte Netztests und Speedtests-Apps genauer untersucht. In Kapitel 3 sind die statistischen Untersuchungen beschrieben. Darauf folgt die Abklärung der Lizenzen sowie die Beschreibung der verwendeten Datenquellen. Das anschließende Kapitel Software Engineering gibt eine Übersicht über den erstellten Server-Code. Das Kapitel 6 befasst sich mit den wichtigsten Punkten der Realisierung. Im darauffolgenden Kapitel werden Arbeiten beschrieben, welche auf den Inhalt dieser Arbeit aufsetzen, bei denen aber keine Zeit vorhanden war, um sie umzusetzen. Der Hauptteil der Dokumentation endet mit dem Kapitel Schlussfolgerung.

## Kapitel 2

### Analyse

#### 2.1 Aussagekraft von Netztests

Wenn die Werbung der unterschiedlichen Mobile Provider angeschaut wird, fällt auf, dass diese mit Labeln wie „Testsieger“ oder „sehr gut“ auftreten. Wie vertrauenswürdig solche Aussagen sind, wird in diesem Abschnitt genauer betrachtet.



(a) connect-Auszeichnung für Salt [3]



(b) Ookla-Auszeichnung für Sunrise [4]



(c) CHIP-Auszeichnung für Swisscom [5]

Abbildung 2.1: Verwendete Auszeichnungen der unterschiedlichen Mobile Provider

Bei den derzeit wichtigsten, für Werbezwecke eingesetzte Netztests in der Schweiz, handelt es sich um folgende:

- connect Mobilfunk-Netztest 2017 [6] [7]
- CHIP Handy-Netztest 2017 [8]
- Ookla Award 2016 [9]
- BILANZ Telekom-Rating 2016 [10]

Beim Betrachten der unten aufgeführten Ranglisten dieser Netztests fällt auf, dass die Rangierungen teilweise sehr unterschiedlich ausfallen. So erreicht Swisscom beim CHIP Handy-Netztest den ersten Rang, liegt beim BILANZ Telekom-Rating aber auf dem zweitletzten Platz. Salt besetzt beim Ookla Award den zweiten Platz vor Swisscom, fällt aber bei drei anderen Netztests durchgehend hinter diesen Provider.

ANBIETER	Schweiz				
		Sunrise	Swisscom	Salt	
<b>SPRACHE</b>	max. 400 Punkte	390	380	352	
<b>DATEN</b>	max. 600 Punkte	561	553	526	
Großstädte	Drivetest	270	94%	92%	88%
Großstädte	Walktest	90	93%	93%	88%
Kleinstädte	Drivetest	120	93%	93%	88%
Verbindungsstraßen	Drivetest	75	94%	92%	87%
Bahn	Walktest	45	90%	93%	87%
<b>SUMME</b>	max. 1000 Punkte	951	933	878	
<b>connect-Urteil</b>		überragend	sehr gut	sehr gut	

Abbildung 2.2: Auszug aus den Ergebnissen des connect Mobilfunk-Netztest 2017 [11]

<b>Gesamtwertung</b>	<b>Swisscom</b>	<b>Sunrise</b>	<b>Salt</b>
<b>Gesamt-Note</b>	<b>1,4 (92,5 Punkte)</b>	<b>1,4 (91,6 Punkte)</b>	<b>2,0 (83,2 Punkte)</b>
<b>Telefonie (40 Prozent)</b>	93,3 Punkte	92,2 Punkte	81,8 Punkte
<b>Mobiles Internet (60 Prozent)</b>	91,9 Punkte	91,3 Punkte	84,0 Punkte
<b>MOBILES INTERNET</b>			
<b>Stadt (zu Fuß)</b>			
<b>Erfolgsquote Datei-Download</b>	99,5%	99,5%	98,4%
<b>Mittlere Downloadrate 1-Thread</b>	54,0 MBit/s	47,8 MBit/s	23,9 MBit/s
<b>Mittlere Downloadrate (4-Threads)</b>	65,0 MBit/s	50,7 MBit/s	31,3 MBit/s
<b>Erfolgsquote Datei-Upload</b>	99,6%	99,2%	97,4%
<b>Mittlere Uploadrate</b>	29,4 MBit/s	29,9 MBit/s	18,7 MBit/s
<b>Erfolgsquote Test-Webseiten aufrufen</b>	99,1%	99,6%	99,1%
<b>Test-Webseiten aufrufen (Durchschnitt)</b>	2,63s	2,66s	2,91s
<b>YouTube Anfangsverzögerung</b>	3,05s	3,15s	3,50s
<b>YouTube-Videoqualität</b>	3,72	3,7	3,65
<b>erfolgreich abgespielte YouTube-Streams</b>	97,9%	96,5%	96,1%
<b>Erfolgsquote App-Tests (Facebook, Dropbox)</b>	99,6%	99,5%	98,8%
<b>App-Tests durchführen (Durchschnitt)</b>	2,84s	3,14s	3,73s

Abbildung 2.3: Auszug aus den Ergebnissen des CHIP Handy-Netztest 2017 [8]

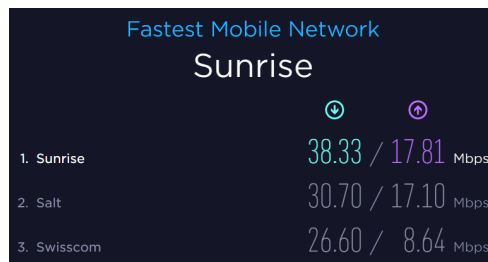


Abbildung 2.4: Ergebnis des Ookla Award 2016 [12]

Rang		Anbieter	Punkte					
2016	2015		Total	Qualität	Innovation	Preis	Flexibilität	Support
<b>Mobiltelefonie</b>								
1	2	Sunrise	22,5	4,3	4,0	4,9	4,7	4,6
2	5	M-Budget	22,4	4,3	4,1	4,7	4,7	4,6
3	1	TalkTalk	22,3	4,1	4,1	4,9	4,6	4,5
4	4	Aldi	21,7	4,2	3,7	5,0	4,7	4,2
5	8	Yallo	21,7	4,2	3,5	5,0	4,6	4,3
6	3	UPC	21,2	4,2	3,9	4,2	4,4	4,5
7	7	Coop Mobile	19,9	3,7	3,2	4,9	4,3	3,9
8	6	Swisscom	19,9	3,7	3,6	4,6	4,1	3,9
9	9	Salt	19,1	3,6	3,6	4,0	3,9	3,9

Abbildung 2.5: Auszug aus den Ergebnissen des BILANZ Telekom-Rating 2016[10]

Ein Grund dafür ist die Art und Weise wie bewertet wurde. Als wichtigster Punkt ist zu beachten, dass BILANZ ausschliesslich die Mobiltelefonie untersuchte, wohingegen connect und CHIP zusätzlich die erreichte Datenrate des mobilen Internets massen. Ookla bewertete sogar nur die erreichten Datenraten. Dies alleine erklärt allerdings die unterschiedlichen Ergebnisse noch nicht, denn wenn man z.B. das Ergebnis der Swisscom beim connect Mobilfunk-Netztest sowie beim Bilanz Telekom-Rating vergleicht, erreicht diese bei connect auch bei der alleinigen Betrachtung der Sprach-Qualität einen guten zweiten Platz, wohingegen sie bei der Bilanz nur den achten Rang erhält.

Deshalb sind ebenfalls die Messmethoden zu beachten, welche sich grundsätzlich voneinander unterscheiden:

- CHIP und connect schickten Autos mit mobilen Messstationen auf die Strasse, massen die Netz-Performance in Zügen und liefen bewusst bestimmte Strecken ab, was einem Drive- und Walk-Test entspricht.
- BILANZ hingegen benutzte die Umfrage als Erhebungstechnik. Sie befragte 50'000 Privat- und 20'000 Firmenkunden zu ihrem Telekommunikationsanbieter in den Bereichen Qualität, Innovation, Preis, Flexibilität und Kundendienst.

- Bei der Messmethode von Ookla handelt es sich um einen sogenannten „crowd-sourced-Ansatz“. Dabei wird eine Speedtest-Applikation zur Verfügung gestellt und sämtliche Messungen, welche die Benutzer dieser App durchführen, werden für die Auswertung benutzt.

### **Nachteile der verschiedenen Messmethoden**

- Drive- und Walk-Test: Dabei handelt es sich um die systematischste der genannten Messmethoden. Wenn diese Tests allerdings falsch durchgeführt werden, kann es zu Fehlaussagen kommen. Beispiele, wie Ergebnisse verfälscht werden können sind:
  - Wenn die Messtrecken unausgewogen sind und es deshalb zu Untergewichtungen einzelner Regionen kommt.
  - Wenn die Teststrecken den Providern im Vorhinein bekannt gegeben werden, damit diese auf den Teststrecken besondere Konfigurationen vornehmen können.
  - Wenn nur die modernsten Mobiltelefone für die Messungen verwendet werden, welche die neusten Standards unterstützen. Dabei wird der aktuelle Stand der Technik der Bevölkerung nicht widerspiegelt.
- Umfrage: Diese Messmethode beinhaltet zwar mehr Aspekte als die Messung der Geschwindigkeit, sie beinhaltet allerdings auch subjektive Ansichten der befragten Personen. Dies wird ersichtlich, wenn man bei den Resultaten des BILANZ Telekom-Rating beachtet, dass M-Budget ebenfalls über das Netz von Swisscom läuft [13], aber durch die Befragten, in Bezug auf die Qualität, viel besser bewertet wurde.
- Crowd-sourced-Ansatz: Es kann sein, dass Benutzer nur genau dann Messungen durchführen, wenn sie das Gefühl haben, dass mit ihrem Netz etwas nicht stimmt. Es könnte also durchaus der Fall sein, dass Netzprobleme zu einem negativen Ausschlag nach unten führen. Andererseits besteht die Gefahr, dass Mobile Provider massenweise Tests in sehr guten Mobilfunkzellen durchführen lassen, um ihr eigenes Ergebnis zu verbessern.

Ein weiterer Grund für die unterschiedlich ausfallenden Rangierungen liegt im gewählten Messgebiet der Tests. So basiert der BILANZ-Netztest auf einer hoffentlich breiten und zufälligen Verteilung der befragten Personen. Die genaue Streuung der Fragebögen über die Schweiz hinweg wird jedoch nicht bekanntgegeben. [10] Auch beim Ookla Award wird die genaue Verteilung der Messungen nicht bekanntgegeben, sondern lediglich ausgewiesen, dass die Provider eine Mindestschwelle von unterschiedlichen messenden Geräten erreichen müssen, um für eine Region beachtet zu werden. [14]

Die Routen von connect und CHIP hingegen, führten mehrheitlich durch Städte oder grosse Orte, wodurch die Netzabdeckung von ländlicheren Regionen wenig Beachtung fand. Dies ist auf der unten aufgeführten Karte ersichtlich, in welcher die Test-Routen von connect eingezeichnet sind. Die Test-Strecken von CHIP umfassen einige Strecken mehr, weichen aber nicht sonderlich davon ab. [15]



Abbildung 2.6: Test-Routen Schweiz des connect-Netztests 2017 [16]

Schliesslich ist zu erwähnen, dass die jeweiligen Unterschiede zwischen den ersten beiden Rängen teilweise verschwindend klein sind, die Telekomanbieter bei einem Sieg aber gross damit werben. Betrachtet man die Ergebnisse des BILANZ Telekom-Rating (siehe Abbildung 2.5), so stellt man fest, dass M-Budget und TalkTalk nur 0,1 Punkte (0,44%) bzw. 0,2 Punkte (0,89%) hinter Sunrise liegen. Gleich enge Ergebnisse sind auch beim connect-Test (siehe Abbildung 2.2) vorzufinden. Bezogen auf die Punktzahl des ersten Rangs von Sunrise, hat Swisscom auf dem zweiten Rang einen Rückstand von lediglich 1,89%. Würde die Skalierung statt über 1'000 Punkte nur über 100 Punkte reichen, so wären die geringen Punkt-Unterschiede eher sichtbar. Ob die Rangierung aufgrund solch knapper Ergebnisse noch gerechtfertigt ist, wird stark bezweifelt.

### Schlussfolgerung

Wenn jemand seinen Anbieter aufgrund solcher Netztests wechseln will, sollte er also genau hinschauen und sich auch fragen, wie wichtig ihm die einzelnen Messkriterien sind. Denn wenn nur genügend unterschiedliche Netztests durchgeführt werden, erreicht jeder Provider einmal den ersten Platz.

Zusammenfassend kann gesagt werden, dass Werbung mit dem Ergebnis aus einem Netztest immer hinterfragt werden soll, da unterschiedlichste Messverfahren angewendet werden und die Gesamtergebnisse nur geringe Unterschiede aufweisen.

## 2.2 Auswertung zu Speedtest-Apps

Es wurden mehrere Speedtest- und Monitoring-Apps ausgewählt, untersucht und verglichen. Bei Speedtest-Apps werden Test-Messungen durch den Klick des Benutzers ausgelöst. Eine Monitoring-App hingegen, läuft permanent im Hintergrund und misst die Aktivitäten des Benutzers.

Für den App-Vergleich wurden folgende Vergleichskriterien festgelegt:

- **Generelle Informationen**  
In dieser Kategorie wurden Informationen zum Hersteller, der Finanzierungsart oder die Anzahl der Downloads zusammengetragen.
- **Benötigte Berechtigungen**  
Hier wurden sämtliche Berechtigungen aufgeführt, die eine App für ihre Ausführung auf einem Android-Gerät benötigt.
- **Verfügbarkeit der Daten**  
In diesem Bereich wurde ausgewertet, wie die Daten dem Benutzer zur Verfügung gestellt werden und ob er Messungen oder Auswertungen von anderen Benutzern sieht.
- **Referenzsysteme**  
Hier wurde die Anzahl und die Standorte der Referenzserver festgehalten. Weiter wurde untersucht, ob diese durch den Benutzer selbst gewählt werden können.
- **Gemessene Daten**  
In diesem Abschnitt wurde ausgewertet, welche Daten pro App gemessen werden. Darunter fallen zum Beispiel Upload- oder Download-Geschwindigkeiten sowie die Antwortzeit.
- **Dargestellte Elemente der gemessenen Daten**  
Es wurde ausgewertet, wie und welche Daten der Messungen ein Benutzer bei der Auswertung des Speedtests angezeigt bekommt.
- **Mögliche Einstellungen durch den Benutzer**  
Hier wurde festgehalten, ob der Benutzer Elemente wie den Referenzserver oder die maximale Datenmenge pro Test selber einstellen kann.
- **Bedienbarkeit**  
Unter diesem Punkt wurde unter anderem beurteilt, ob die Navigation innerhalb der App verständlich ist oder ob Erklärungen zu den gemessenen Daten vorhanden sind.
- **Besonderheiten von Monitoring Apps**  
Da sich unter den ausgewerteten Apps auch Monitoring-Apps befanden, welche zusätzliche Funktionalitäten anbieten, wurde diese Kategorie erstellt. Darin wurde der Funktionsumfang untersucht.

Bei den untersuchten Apps handelt es sich um die folgenden, auf welche in den nächsten Abschnitten genauer eingegangen wird:

<b>Name der App</b>	<b>Unternehmen</b>	<b>Grund für die Auswahl</b>
cnlab Speed Test [17]	cnlab AG [18]	Die Messdaten für die im Kapitel 6 „Realisierung“ umgesetzte Logik stammen aus dieser App.
Speedtest.net [19]	Ookla [20]	Der Anbieter Ookla macht mit seinen gross durchgeführten Netztests auf sich aufmerksam. Sunrise wirbt als Testsieger mit diesem Netztest. Zudem erreicht die App alleine in Androids Play Store 50'000'000 bis 100'000'000 Downloads, was alle anderen Speedtest-Apps bei weitem übersteigt.
CHIP Netztest [21]	CHIP Digital GmbH [22]	Diese App wurde ausgewählt, da auch CHIP ein grosser Netztest-Anbieter ist. Bei diesem Netztest ging Swisscom als Sieger hervor, was nun durch Swisscom beworben wird.
Geschwindigkeits-Test 3G & 4G [23]	OpenSignal.com [24]	Die App Geschwindigkeits-Test 3G & 4G wurde ausgewählt, da sie viele Ansichten und Auswertungen anbietet.
4G map & speedtest [25]	Sensorly.com [26]	Die App hat im Google Play Store einen guten Eindruck gemacht und wirkt sauber aufbereitet. Deshalb wurde sie in diesen Vergleich miteinbezogen.
Speed Test & QoS 3G 4G WiFi [27]	nPerf.com [28]	Hierbei handelt es sich um die App eines Unternehmens, welches sich stark im Bereich der Netztests einsetzt.
Traffic Monitor+ mit Speedtest [29]	RadioOpt GmbH [30] [31]	Der Grund für die Auswahl dieser App ist, dass sie als Monitoring-App das erste Ergebnis im Google Play Store ist und sie einen soliden Eindruck macht.
U get [32]	P3 insight GmbH [33]	Das Unternehmen P3 spielt bei den Netztests von connect eine erwähnenswerte Rolle [34]. Es wurde deshalb entschieden ihre App genauer anzuschauen.

Die nachfolgenden Speedtest-Apps wurden kurz angeschaut, aber aus verschiedenen Gründen nicht genauer untersucht:

- Der „Cisco Data Meter“ [35] funktionierte auf beiden Android-Testgeräten nicht. Auch im Google Play Store wurden solche negative Rückmeldungen gefunden, welche ähnliche Probleme beschreiben.
- Der „FLOQ Netztest (Speedtest)“ [36] wurde ausgeschlossen, da es sich dabei lediglich um die iOS-Version des „CHIP Netztests“ [21] handelt, welche zu den untersuchten Apps gehört. Zum Zeitpunkt dieser Bachelorarbeit, befand sich die iOS-App in Überarbeitung und konnte nicht heruntergeladen werden. [37]
- Der „Network Master - Speed Test“ [38] wurde nicht untersucht, da mit dieser App, trotz des Namens, keine Speedtests durchgeführt werden können. Der Zweck der App liegt darin, die Netzgeschwindigkeit des Gerätes optimaler zu nutzen.
- Die App „Speed Test“ [39] wurde nicht gewählt, da die Darstellung der Messungen schlecht umgesetzt wurde.
- Die App „Einfacher Speed Test [40]“ besitzt schöne Darstellungselemente, jedoch fehlt der Funktionsumfang.

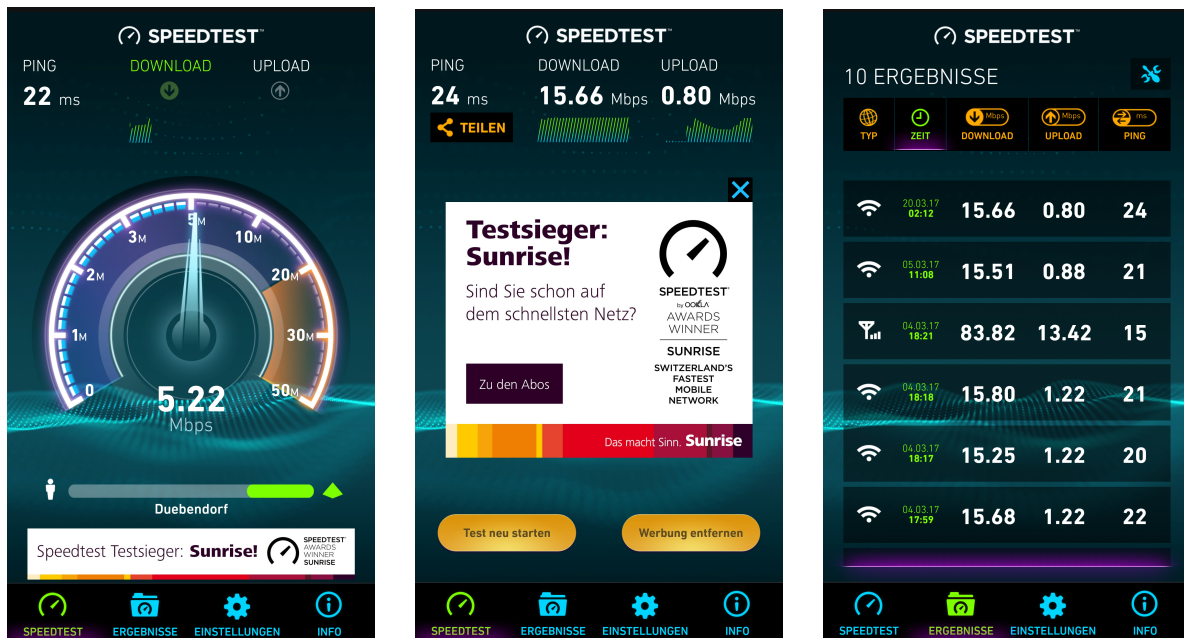
### **2.2.1 cnlab Speed Test [17]**

Bei der cnlab Speed Test-App handelt es sich um den Ausgangspunkt für diesen Vergleich. Der Nutzen des Benutzers liegt darin, dass er seine Messungen bis ins Detail anschauen kann. Um die erfassten Daten auszuwerten, ist allerdings auch ein gewisses Fachwissen erforderlich. Weitere Details wie die App verbessert werden könnte und was bereits gut umgesetzt wurde, sind im Abschnitt 2.3 „Auswertung zur cnlab Speed Test App“ ersichtlich.

## 2.2.2 Speedtest.net [19]

Bei Speedtest.net handelt es sich um eine reine Speedtest-App, es sind also keine Monitoring Funktionalitäten vorhanden. Ganz generell werden nur wenige Funktionen angeboten, was zu einer einfachen Navigation und Benutzung führt. In dieser App wird In-App Werbung geschaltet. Zwar ist somit eine der Einnahmequellen des Apps bekannt, allerdings wird aufgrund der geschalteten Werbung die Objektivität der Tests hinterfragt (siehe dazu Abbildung 2.7b).

Die gemessenen Daten werden nicht genauer erklärt und bei den erhaltenen Resultaten wird nicht angegeben, ob es sich dabei um eine Minimal-, Maximal- oder eine Durchschnittsangabe handelt. Der Benutzer kann in der App seine eigenen Messungen anschauen, diese aber nicht mit den Ergebnissen von anderen Benutzern vergleichen. Auf der Website des Anbieters Ookla [41] ist es allerdings möglich, Auswertungen zu unterschiedlichsten Länder anzuschauen.



(a) Laufende Messung

(b) Gezeigte Werbung

(c) Ergebnisliste

Abbildung 2.7: Screenshots der Speedtest.net-App

### 2.2.3 CHIP Netztest [21]

Es handelt sich beim CHIP Netztest um eine Speedtest-App mit Zusatzfunktionen. Nebst der eigentlichen Performance-Messung kann der Benutzer auch Probleme mit seinem Netzwerk melden oder eine subjektive Netzbewertung ausfüllen.

Der Benutzer erhält nach der Durchführung des Speedtests eine Bewertung, welche ihm seine Verbindungsqualität im Vergleich zu allen anderen Testern anzeigt. Es wird angegeben, dass man sich, auf der Seite des Herstellers FLOQ, Karten mit der Netzabdeckung anschauen kann. Diese Website [42] war allerdings bis zum Ende dieser Auswertung zu keinem Zeitpunkt erreichbar, da sie im Umbau ist. Der Frequently Asked Questions (FAQ)-Bereich innerhalb der App wiederum, bietet unerfahrenen Benutzer eine gute Unterstützung zu den erhaltenen Resultaten.

Die App enthält im Allgemeinen etwas zu viel Text und wirkt durch ihre Farbwahl etwas blass und unfreundlich.

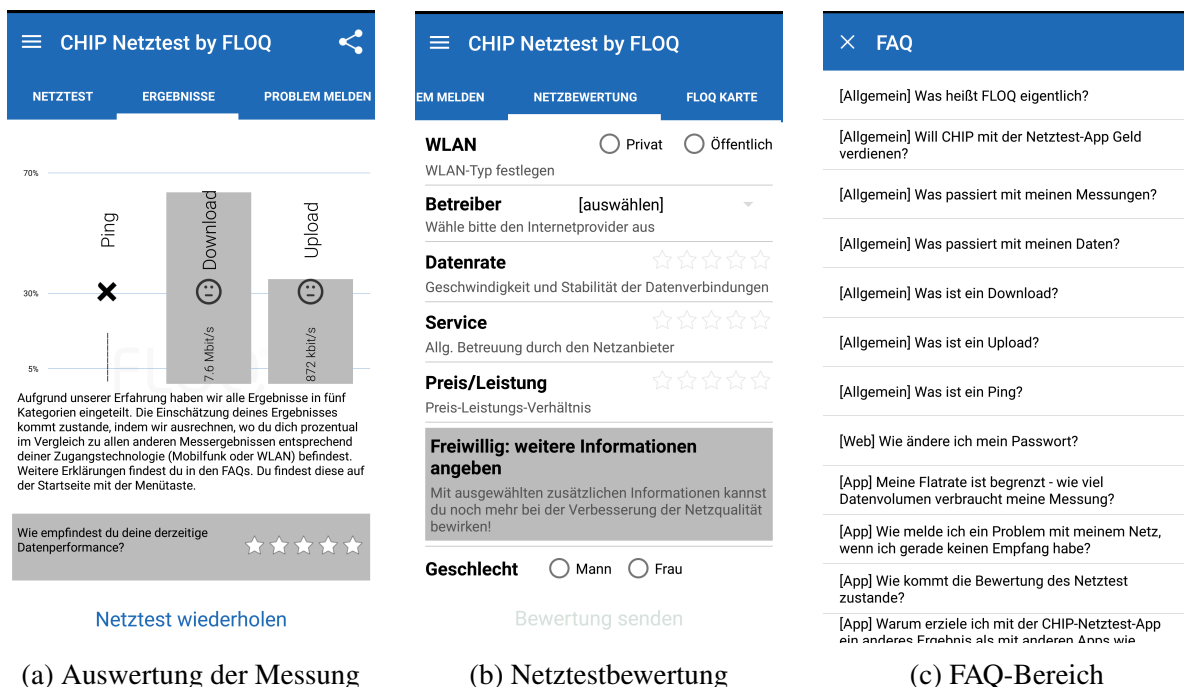


Abbildung 2.8: Screenshots der CHIP Netztest-App

## 2.2.4 Geschwindigkeits-Test 3G & 4G [23]

Bei dieser App handelt es sich um eine Speedtest- und Monitoring-App. Beim ersten Starten müssen die Allgemeine Geschäftsbedingungen (AGB) akzeptiert werden. Es konnten aber keine speziellen Punkte innerhalb der AGB festgestellt werden. Da es sich hierbei um eine App von einem amerikanischen Hersteller handelt, wird vermutet, dass aufgrund der anderen Rechtslage eine AGB üblich ist. [43]

Der Benutzer dieser App erfährt seine eigene Verbindungsqualität sowie die Netzabdeckung und Messdaten verschiedener Provider in seiner Region. Auf einer Karte kann für jeden Ort herausgefunden werden, welcher Anbieter im Durchschnitt am schnellsten ist. Zudem werden Provider-Antennen und WLANs in der Nähe des Benutzers angezeigt. Dabei handelt es sich um öffentliche WLANs wie zum Beispiel jene der SBB an den Bahnhöfen. Bei den meisten angezeigten WLANs handelt es sich allerdings um UPC Wi-Free-Netzwerke, welche ausschliesslich UPC-Kunden zur Verfügung stehen [44]. Die einzige Monitoring-Funktion prüft, zu wie viel Prozent der Zeit man sich in welchem Netz (4G/3G/2G/Kein Signal) befand.

Die Bedienung ist teilweise nicht intuitiv, da zum Beispiel die eigenen Messungen als erstes auf einer Karte dargestellt und nicht in einer Liste aufgeführt werden.

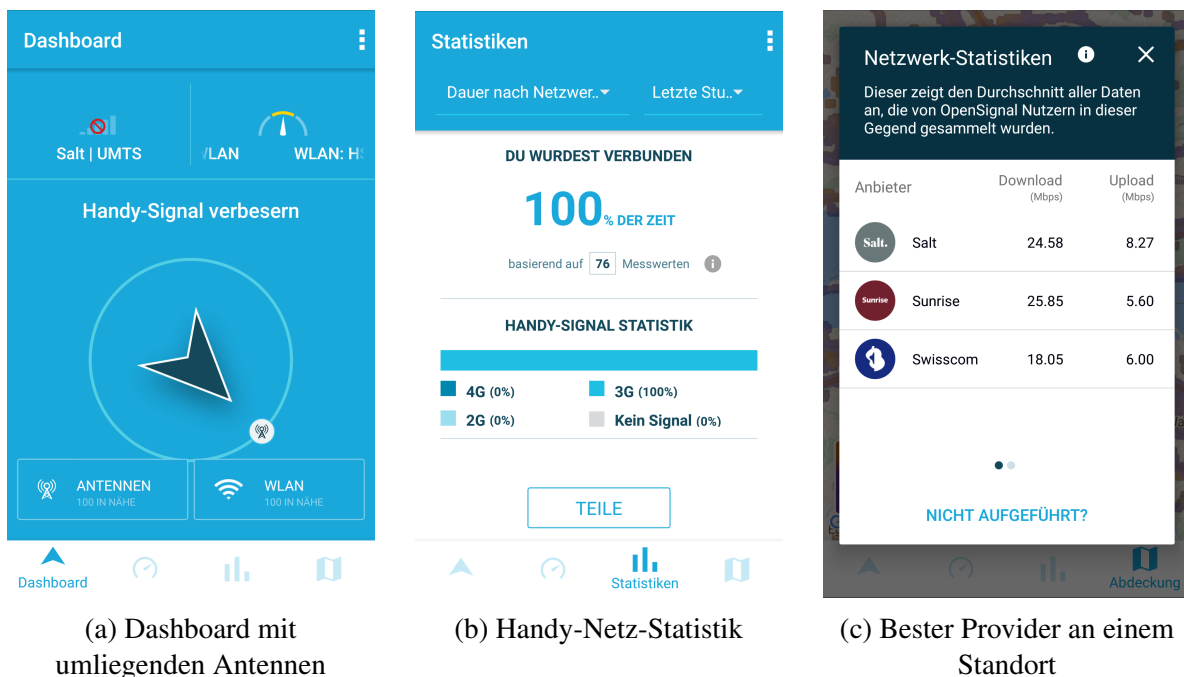


Abbildung 2.9: Screenshots der Geschwindigkeits-Test 3G & 4G-App

## 2.2.5 4G map & speedtest [25]

Es handelt sich um eine reine Speedtest-App. Beim ersten Start müssen die AGB akzeptiert werden. Bei diesem Schritt stimmt man zu, dass anonymisierte Daten aufgezeichnet und vom Hersteller sensorly ausgewertet werden dürfen.

Der Benutzer erhält die Informationen zu seiner Verbindungsqualität in einer verständlichen Weise dargestellt. Direkt während des Tests wird angegeben, wie lange er für z.B. den Upload eines Bildes einer bestimmten Grösse benötigen würde. Leider sind diese Informationen nach beenden des Tests nicht mehr ersichtlich, es werden lediglich die erreichten Werte abgespeichert. Der Benutzer sieht anhand einer Karte die Netzabdeckung sowie einen Provider-Vergleich in seiner Region. Diese Karten können auch über die Website [45] angeschaut werden. Es werden zusätzlich zu den üblichen Tests (Download, Upload und Ping) auch die Webpage-Ladezeiten von google.com, facebook.com und youtube.com gemessen. Mittels der „Map Trip“-Funktion kann eine längere Messung (z.B. auf dem Weg zur Arbeit) durchgeführt werden.

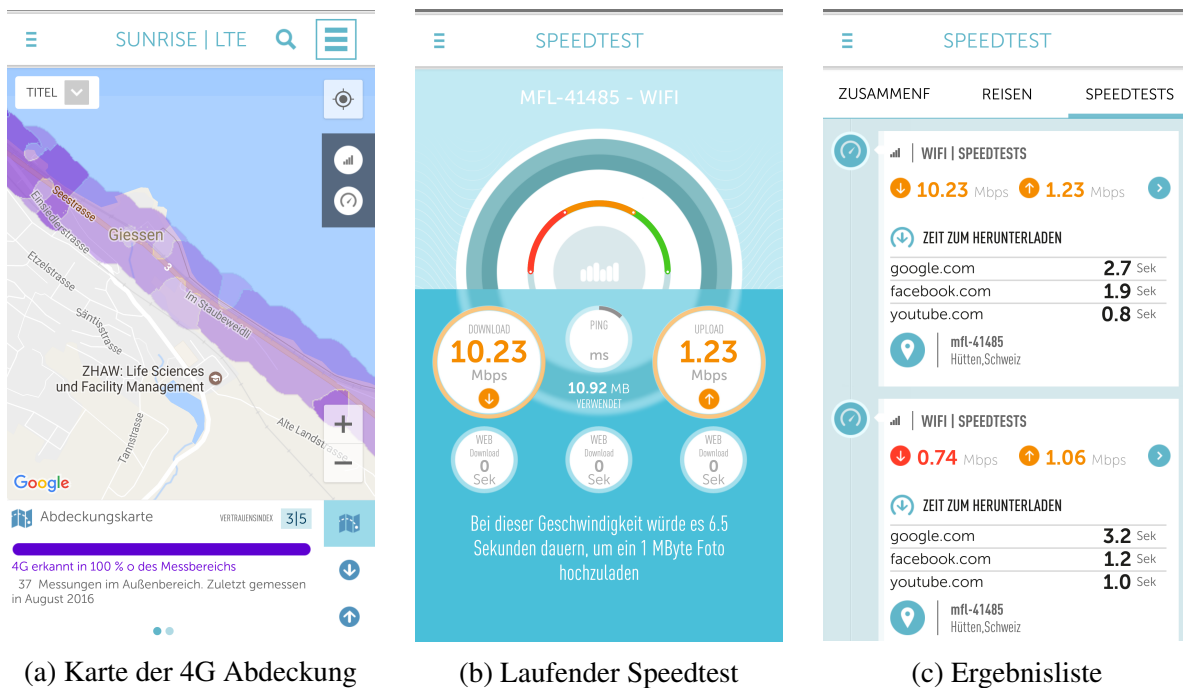


Abbildung 2.10: Screenshots der 4G map & speedtest-App

## 2.2.6 Speed Test & QoS 3G 4G WiFi [27]

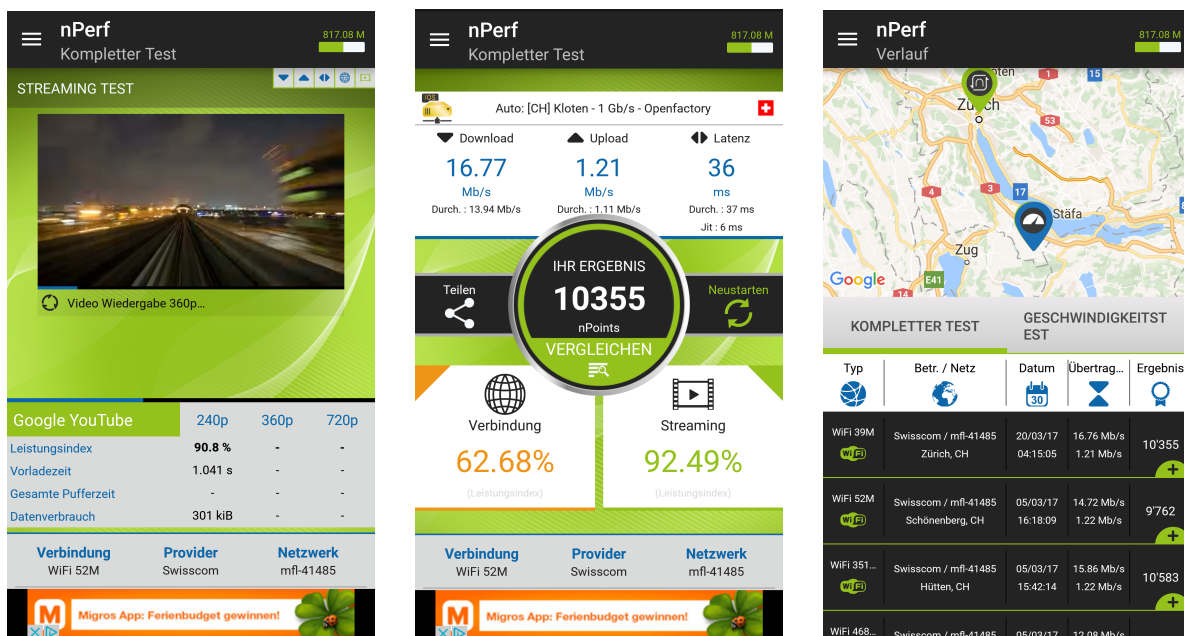
Es handelt sich um eine Speedtest- und Monitoring-App. Die Monitoring-Funktionen stellen allerdings nur einen kleinen Teil der App dar. So wird lediglich angezeigt, wie viele Daten über WLAN oder über das Mobil-Netz insgesamt gesendet und empfangen wurden. In dieser App wird Werbung geschaltet. Es konnte nicht herausgefunden werden, wie viel für die zu schaltende Werbung zu bezahlen ist.

Es wird eine breite Auswahl an Tests angeboten. So kann nebst dem „normalen“ Speedtest auch ein Verbindungstest und ein Streaming-Test durchgeführt werden. Bei ersterem werden Webpage-Ladezeiten gemessen, beim zweiten sind es Youtube-Videos, welche in verschiedenen Qualitäten geladen werden.

Der Benutzer kann nach Beendigung des Komplet-Tests einen Vergleich anschauen, bei welchem das eigene Ergebnis dem nationalen und dem lokalen Durchschnitt gegenübergestellt wird. Allerdings ist die Messeinheit dieses Vergleichs in Punkten (nPts), wodurch die Bewertung nicht genau nachvollzogen werden kann. Als zusätzlichen Vergleich gibt es einen Leistungsindex für den Verbindungs- und den Streaming-Test als Prozentzahl, welche ebenfalls nicht erklärt wird.

In der App können auch die Karten zur Netzabdeckung und zur Empfangsgeschwindigkeit angeschaut werden. Es fehlen leider auch hier sämtliche Erklärungen, was die Aussagekraft der Tests verkleinert. Mittels aktiver Kartographie kann eine längere Messung durchgeführt werden (man muss sich dazu allerdings bewegen, ansonsten wird nicht gemessen). Beim Testen dieser Funktionalität wurde festgestellt, dass nur auf Strassen, nicht im Zug, gemessen werden kann.

Die App wirkt aufgrund ihrer vielen Funktionalitäten und den geschalteten Werbungen überladen. Die Bewertung bei gewissen Tests konnte leider nicht genauer nachvollzogen werden.



(a) Laufender Streaming Test

(b) Auswertung Komplet-Test

(c) Ergebnisliste

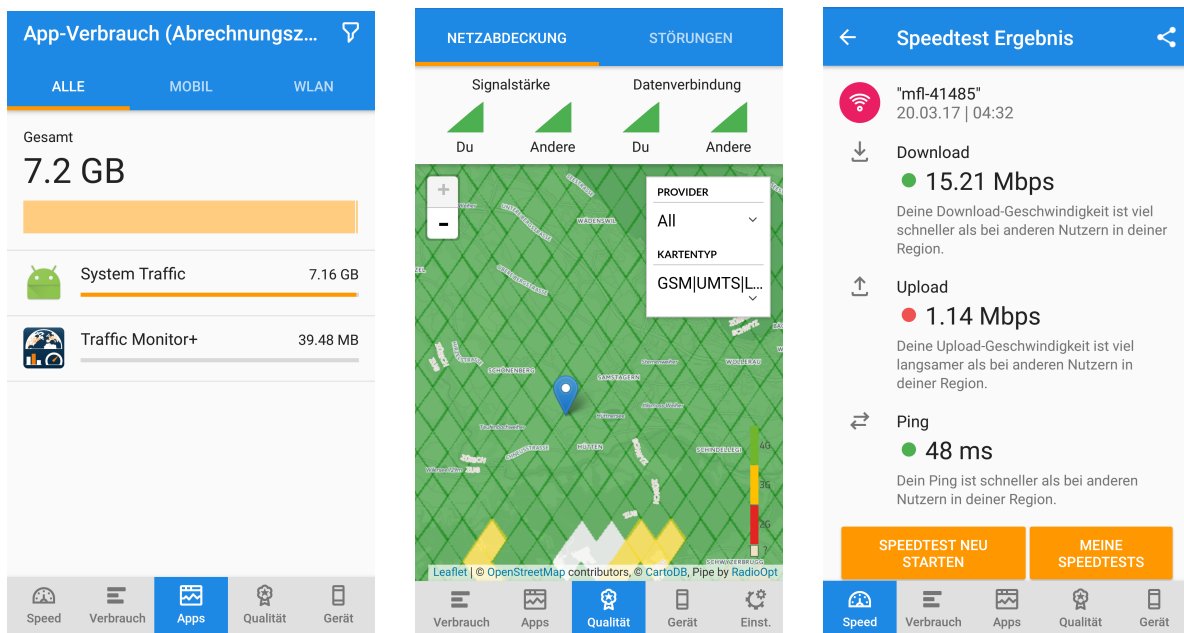
Abbildung 2.11: Screenshots der Speed Test & QoS 3G 4G WiFi-App

## 2.2.7 Traffic Monitor+ mit Speedtest [29]

Nebst den Monitoring-Funktionen können auch Speedtests durchgeführt werden. Beim ersten Start der App muss die Endnutzer-Lizenzvereinbarung bestätigt werden. Darin wird festgehalten, dass anonymisierte Daten gesammelt und ausgewertet werden dürfen.

Der Benutzer erfährt seine Verbindungsgeschwindigkeit und erhält eine Aussage zur Qualität seiner Messung im Vergleich zu anderen in seiner Region. Er sieht die Netzabdeckung in seiner Region und erhält Informationen zu seinem Datenverbrauch. Beim Datenverbrauch pro App wird allerdings nur zwischen System Traffic und dem Traffic Monitor+ unterschieden. Es kann somit nicht herausgefunden werden, welche App wie viel des Datenvolumens benötigt hat.

Die App hat die Erwartungen nicht erfüllt, denn sie kann die von ihr versprochenen Leistung nicht bieten. Beispiel: „Traffic Monitor misst die Datennutzung App-übergreifend sowie für jede App einzeln.“ [29] Wie bereits erwähnt, wurde keine Auswertung zu den einzelnen Apps erstellt. Die Navigation ist nicht immer verständlich und es fehlen die Erklärungen. So konnte zum Beispiel die Karte mit der Netzabdeckung der Region nur schwer interpretiert werden.



(a) Auswertung des Datenverbrauchs

(b) Karte

(c) Ergebnisliste

Abbildung 2.12: Screenshots der Traffic Monitor+ mit Speedtest-App

## 2.2.8 U get [32]

Bei U get handelt es sich um die einzige reine Monitoring-App. Beim ersten Starten müssen die Nutzungs- und Datenschutzbedingungen angenommen werden. Diese beinhalten, dass die Daten anonymisiert gesammelt und weitergegeben werden dürfen. Die App läuft die ganze Zeit im Hintergrund und beobachtet das Nutzerverhalten. Konkret werden die maximale Downloadgeschwindigkeit, die Signalstärke, die Gesprächsabbruchrate, die Technologie-Verteilung (2G/3G/4G/WiFi) und die Datennetzverfügbarkeit beobachtet. Der Benutzer erhält so eine genaue Übersicht zu seiner effektiven Nutzung des Datenvolumens.

Die App kann sich durch gute und benutzergerechte Erklärungen hervorheben. Es wird allerdings nicht genauer erklärt, anhand von welchen Kriterien die Signalstärke als „Ausgezeichnet“, „Gut“, usw. bezeichnet wird.

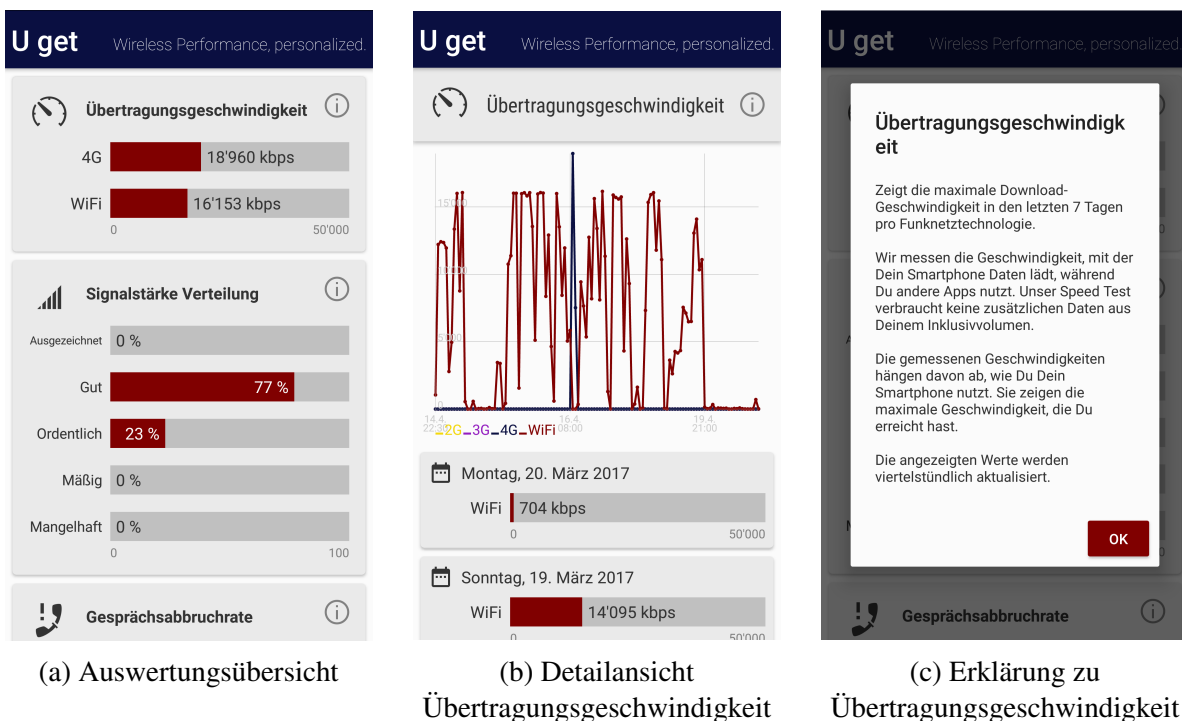


Abbildung 2.13: Screenshots der U get-App

In der unten stehenden Tabelle werden die wichtigsten Erkenntnisse aus der Analyse der einzelnen Apps aufgeführt:

<b>Name der App</b>	<b>Art der App</b>	<b>Sind AGB zu akzeptieren?</b>	<b>Anzahl Berechtigungen</b>	<b>Finanzierungsart</b>
cnlab Speed Test [17]	Speedtest	Nein	11	Nutzung der gesammelten Daten für Engineering- und Beratungsprojekte.
Speedtest.net [19]	Speedtest	Nein	7	Es wird Werbung in der App geschaltet. Eine werbefreie Version ist für CHF 1.00 erhältlich. Das Daten- und Analysetool „Speedtest Intelligence“ wird verkauft. [46]
CHIP Netztest [21]	Speedtest	Nein	15	CHIP sammelt anonymisierte Daten für Messung der Netzqualität. Dafür bezahlen die Netzbetreiber. [47]
Geschwindigkeits-Test 3G & 4G [23]	Speedtest und Monitoring	Ja	17	Weiterverkauf der anonymisierten Messdaten [48]
4G map & speedtest [25]	Speedtest	Ja	15	Weiterverkauf der Messdaten [49]
Speed Test & QoS 3G 4G WiFi [27]	Speedtest und Monitoring	Nein	13	Weiterverkauf der anonymisierten Daten. [50] In der App wird Werbung geschaltet. Ein werbefreies App mit zusätzlichen Elementen (unbegrenzter Verlauf, Echtzeit-Lokalisierung und Netzwerkinformationen) ist für CHF 2.49 pro Jahr erhältlich.
Traffic Monitor+ mit Speedtest [29]	Speedtest und Monitoring	Ja	17	Unbekannt, eventuell Weiterverkauf der Daten
U get [32]	Monitoring	Ja	12	Unbekannt, eventuell Weiterverkauf der Daten

Der detaillierte Vergleich ist im Abgabeordner unter „1\_Analyse\2\_Speedtest-Apps \Vergleich\_Speedtest\_Apps.xlsx“ aufzufinden. Wie auf den Abgabeordner zugegriffen werden kann, ist im Kapitel 11.1 erklärt. Die Bewertung der Vergleichs-Daten wird durch die cnlab AG fertiggestellt, da den Studenten das Fachwissen dazu fehlt.

## 2.3 Auswertung zur cnlab Speed Test-App

Aufgrund der cnlab Speed Test-App wurde dieser Vergleich erstellt. Es wird nachfolgend, anhand der Erkenntnisse aus dem App-Vergleich, festgehalten, was bei dieser App bereits gut umgesetzt wurde und wo es noch Verbesserungspotenzial gibt.

### 2.3.1 Stärken

- **Vertrauen**  
Die App kommt im Vergleich zu den oben aufgeführten App mit einer kleinen Anzahl von Berechtigungen aus. Zudem sind in der Beschreibung im Google Play Store Erklärungen zu den benötigten Berechtigungen vorhanden. Ebenfalls ist ein Abschnitt dem Datenschutz und der Firma cnlab selbst gewidmet, was für zusätzliches Vertrauen sorgt.
- **Einstellungsmöglichkeiten**  
Der erfahrene Benutzer kann in der App viele Einstellungen vornehmen. So kann zum Beispiel der Referenzserver oder die Messdauer selbst bestimmt werden.
- **Detaillierte Auswertungen**  
Nach einem durchgeführten Speedtest gibt die App detaillierte Resultate zu den gesammelten Daten. Es wird überall angegeben, ob es sich um Durchschnitts- oder Maximal-Werte handelt.

### 2.3.2 Verbesserungspotential

- **Fehlende Erklärungen**  
Wie oben erwähnt, werden detaillierte Messresultate zurückgegeben. Dies kann für unerfahrene Benutzer allerdings schnell überfordernd sein. Es wird deshalb vorgeschlagen, dass die gemessenen Elementen mit Erklärungen ergänzt werden.
- **Verständliche Vergleiche**  
Die App könnte grundsätzlich mit mehr Feedback für den Benutzer ausgestattet werden. Möglich wäre zum Beispiel, nach der Messung anzuzeigen, wie lange man mit der erreichten Download-Geschwindigkeit für einen bestimmten Datei-Download benötigen würde. Dies wurde bei „4G map & speedtest“ (siehe Abschnitt 2.2.5) so umgesetzt.  
Dem Benutzer könnte weiter angezeigt werden, welche Aktivitäten, wie gut mit der gemessenen Geschwindigkeit erledigt werden können. So zeigt zum Beispiel die App „Einfacher Speedtest[40]“ mit farbigen Symbolen an, wie flüssig Videos gestreamt oder wie schnell E-Mails versendet werden können.
- **Vergleich mit anderen Benutzern**  
Die erreichten Geschwindigkeiten könnten mit Messwerten aller anderer Benutzer verglichen werden. So stellt die App „Speed Test & QoS 3G 4G WiFi“ (siehe Abschnitt 2.2.6) die eigene Geschwindigkeit mit dem lokalen und dem nationalen Durchschnitt ins Verhältnis.

Weiter kann zum Beispiel auf der Karte von „Geschwindigkeits-Test 3G & 4G“ (siehe Abschnitt 2.2.4) direkt nachgeschaut werden, welcher Provider an welchem Punkt die besten Messwerte erzielt.

- **Weitere Test-Arten**  
Bei der App von nPerf (siehe Abschnitt 2.2.6) wurden neben dem „normalen“ Speedtest zusätzliche Tests eingeführt, nämlich solche für die Webpage-Ladezeiten sowie Streaming-Tests. Es wird vorgeschlagen, dass die cnlab-App ebenfalls eine solche Funktionalität bietet, da sie dem Benutzer weitere wertvolle Erkenntnisse liefern kann.
- **Weitere Einstellungsmöglichkeiten**  
In den Einstellungen wäre es sinnvoll, wenn pro Test eine Maximal-Datenmenge festgelegt werden kann. Somit erleben Benutzer mit eingeschränktem Internet-Volumen keine bösen Überraschungen, wenn mehr Daten als erwartet verbraucht wurden.
- **Transparenz**  
Im About zur cnlab-App könnte ein Abschnitt zur Finanzierungsart der App erstellt werden. Die App würde sich durch diese Transparenz klar von den anderen Anbietern abheben, da die Finanzierungsarten dort nicht immer offensichtlich sind.

Falls die erwähnten Verbesserungspunkte umgesetzt werden, kommt die App der cnlab AG einer Ideal-App schon sehr nahe.

# Kapitel 3

## Statistische Auswertungen

### 3.1 Einführung in die Statistik

Bei den Daten, welche statistisch auszuwerten sind, handelt es sich um Speedtest-Messungen aus der cnlab App. Das Ziel ist es, aufzuzeigen, ab wann statistische Auswertungen von Speedtest-Messungen aussagekräftig sind. Bei der cnlab AG besteht bereits die kumulative Auswertung der Daten. Anhand des Beispiels der Download-Datenrate werden die für die weiteren Kapitel benötigten Begriffe hergeleitet.

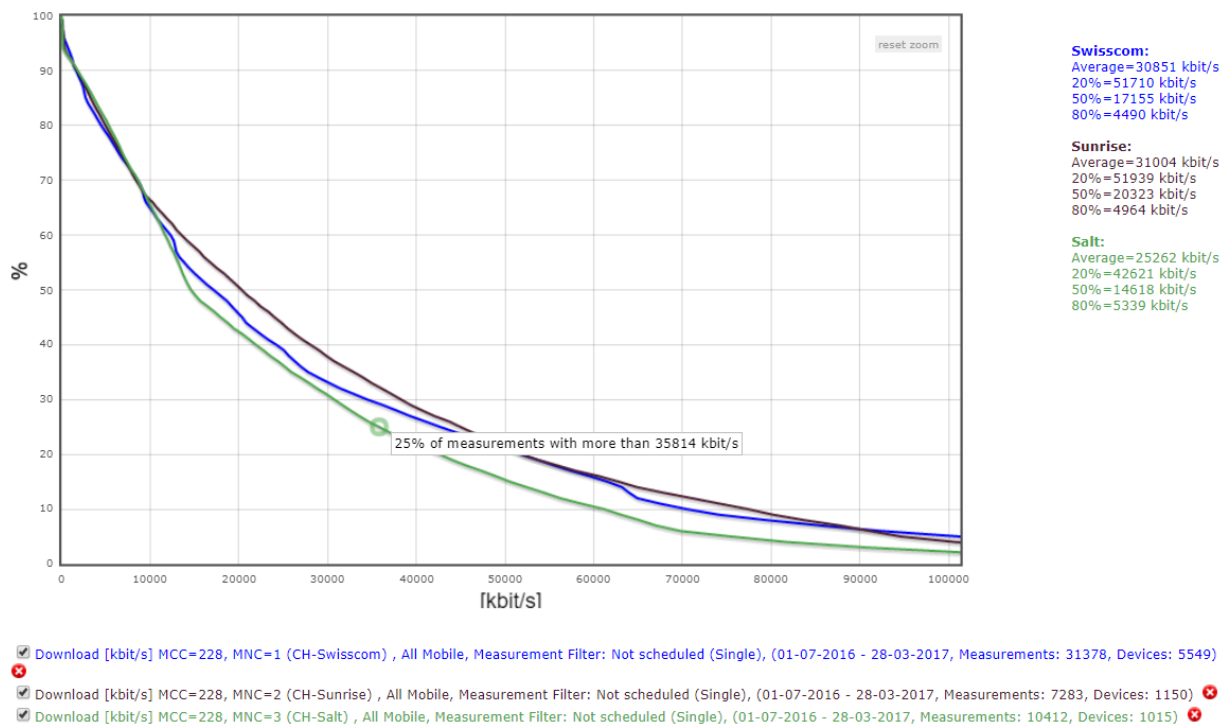


Abbildung 3.1: Kumulative Verteilung der Download-Datenrate im Zeitraum vom 01.07.2016 bis zum 28.03.2017, erstellt mit cnlab Mobile Cockpit [51]

Im oben aufgeführten Bild ist ersichtlich, dass bei Salt 25% aller Messungen eine Download-Datenrate von mehr als 35'814 kbit/s erreicht haben. Umgekehrt heisst das, dass 75% aller Messungen unter dem Wert von 35'814 kbit/s liegen, hier spricht man auch vom 75ten Perzentil.

Wenn vom 25ten Perzentil gesprochen wird heisst das, dass unter diesem Wert die 25 schlechtesten Prozent der gemessenen Daten liegen. Ein spezielles Perzentil ist das 50te, bei dem die Hälfte der gemessenen Werte über diesem Wert und die andere Hälfte darunter liegt. Das 50te Perzentil wird auch Median genannt.

Wenn vom Durchschnitt gesprochen wird, dann entspricht dies allen aufsummierten Messungen, geteilt durch die Anzahl Messungen. Der Durchschnitt hat allerdings das Problem, dass Ausreisser nach oben oder nach unten den Wert stark verändern. Aus diesem Grund wird im weiteren Verlauf dieser Auswertungen nur der Median bzw. das 50te Perzentil verwendet, denn dieser ist nicht anfällig auf solche Ausreisser. Diese Aussage wird im nachfolgenden Bild nochmals aufgezeigt:

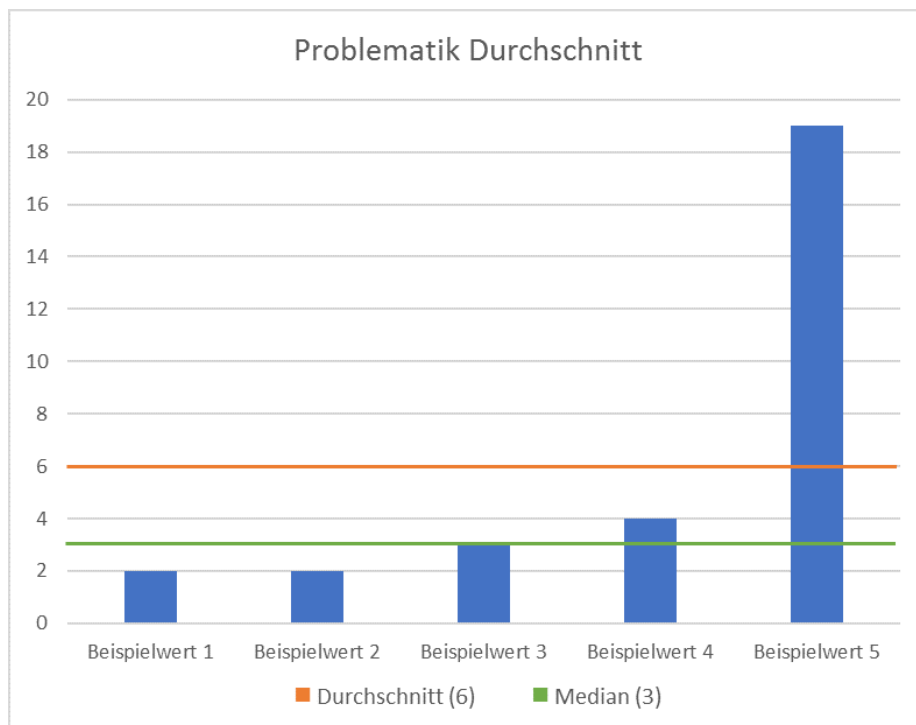


Abbildung 3.2: Problematik des Durchschnitts und Lösung durch die Verwendung des Medians

Der Durchschnitt von 6 ergibt sich aus folgender Rechnung:

$$\frac{2 + 2 + 3 + 4 + 19}{5} = 6$$

Das Problem ist nun, dass vier der fünf Messungen den Durchschnitt gar nie erreichen. Im Gegensatz dazu kommt der Median bei 3 zu liegen, denn dann befinden sich zwei Messungen über diesem Wert und zwei Messungen darunter.

Die oben erwähnten Perzentile werden typischerweise in einem Boxplot dargestellt. Wie ein Boxplot aussieht kann in untenstehender Beispielgrafik nachvollzogen werden:

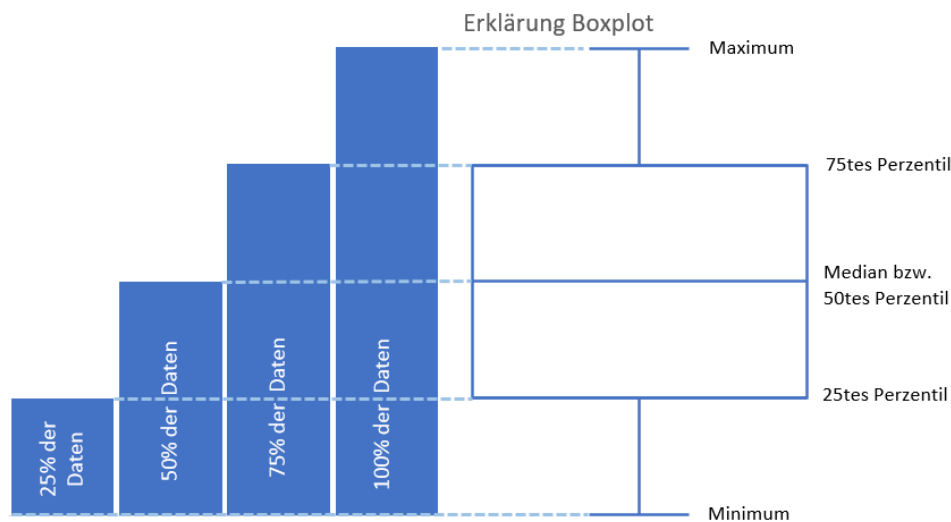


Abbildung 3.3: Erklärung zum Boxplot

### 3.1.1 Datenaufbereitung

Es stehen Daten im Zeitraum vom 01.07.2016 bis zum 28.03.2017 zur Verfügung. Dabei handelt es sich um Total 309'130 einzelne Speedtest-Messungen.

Die schlussendlich verwendeten, validen Daten bestehen aus allen Daten, auf welche der folgende Filter angewendet wird:

- Valid = 1 (Messung wurde als valid eingestuft)
- MCC = 228 (Provider der Schweiz)
- MNC = 1 oder 2 oder 3 (Swisscom, Sunrise, Salt)
- longitude zwischen 5.97 und 10.47 (Messung innerhalb der Schweiz)
- latitude zwischen 45.82 und 47.8 (Messung innerhalb der Schweiz)
- ConnectionTypeIDDownload != 131 oder 157 oder 60 (Nur Mobile-Messungen, kein WLAN)
- Roaming = 0 (In der Schweiz erstellte Messungen, keine VPN-Verbindung)
- UsedForReport = 1 (Messung wurde als valid eingestuft)

- DownloadAvg  $\geq 0$  (Download-Datenrate grösser oder gleich 0)
- DownloadMax  $> 0$  (Download konnte korrekt durchgeführt werden)
- UploadAvg  $\geq 0$  (Upload-Datenrate grösser oder gleich 0)
- RTTAvg  $\geq 0$  (Round Trip Time grösser oder gleich 0)
- ConnectionTypeID  $\neq 131$  oder 157 oder 60 (Nur Mobile-Messungen)
- Scheduler = 0 (Nur Einzelmessungen)
- ConnectionFailure = 0 (Kein Verbindungsfehler aufgetreten)
- SimOperatorid = 2 oder 4 oder 5 (Swisscom, Sunrise, Salt)  
Um z.B. nur Swisscom-Messungen herauszusuchen, wird die Kombination der Filter MNC=1 und SimOperatorid=2 benötigt. Für Sunrise ist die Kombination MNC=2 und SimOperatorid=4 und bei Salt MNC=3 und SimOperatorid=5.

Es resultieren somit 48'306 valide Daten, welche für die Auswertungen in den folgenden Unterkapiteln verwendet werden. Die Auswertungen konzentrieren sich auf die Download-Datenrate. Alle Excel-Dateien mit den hier aufgeführten Berechnungen befinden sich im Abgabeordner unter „2\_Statistische\_Auswertungen“.

Die Abbildung 3.4 stellt die Auswertung der Download-Datenrate über den gesamten Zeitraum dar. Bereits in dieser Darstellung ist ersichtlich, dass es wahrscheinlich grosse Ausreisser in den Daten hat, da das 75te und das 100te Perzentil weit auseinanderliegen. Anhand der zweiten Abbildung 3.5 wird gezeigt, dass durch die Ausreisser der Durchschnitt, welcher in blau eingezeichnet ist, weit über dem 50ten Perzentil liegt. In der Darstellung 3.6 des 95ten Perzentils wird aufgezeigt, dass es tatsächlich die höchsten 5% der Messungen sind, welche zu diesen starken Ausreissern führen. Die Download-Datenrate dieser Messungen sind bis zu 4,5 mal grösser als die restlichen 95% der erreichten Werte. Es kann schon hier festgehalten werden, dass für eine aussagekräftige Auswertung mindestens die obersten 5% der Messungen ausgeschlossen werden sollten.

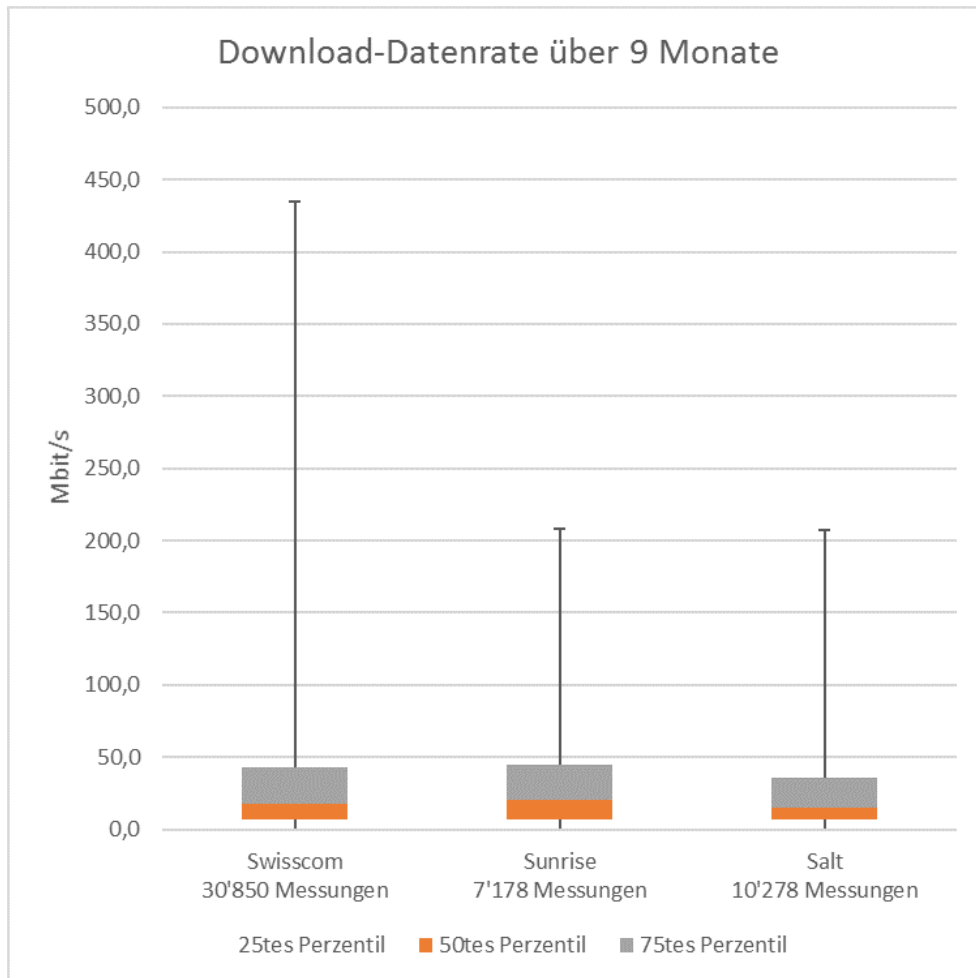


Abbildung 3.4: Download-Datenrate über den gesamten Zeitraum

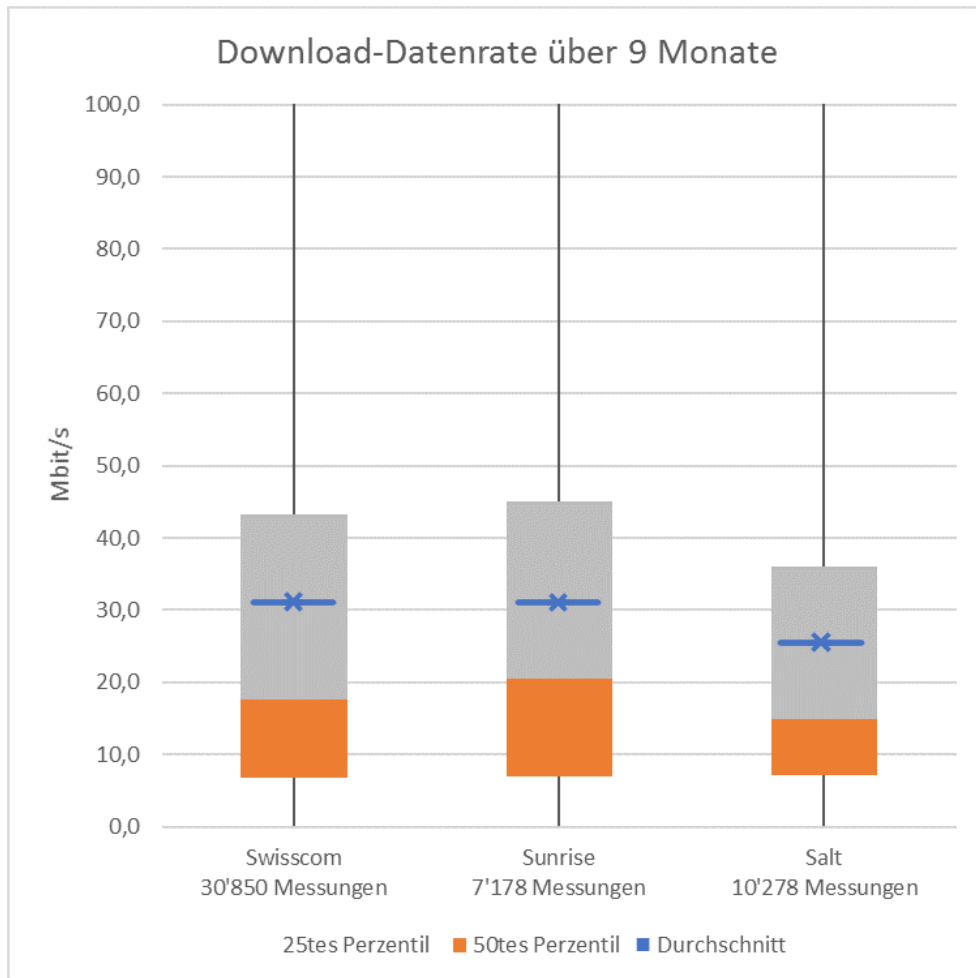


Abbildung 3.5: Hervorhebung des Durchschnitts, welcher aufgrund der Ausreisser weit über dem 50ten Perzentil liegt

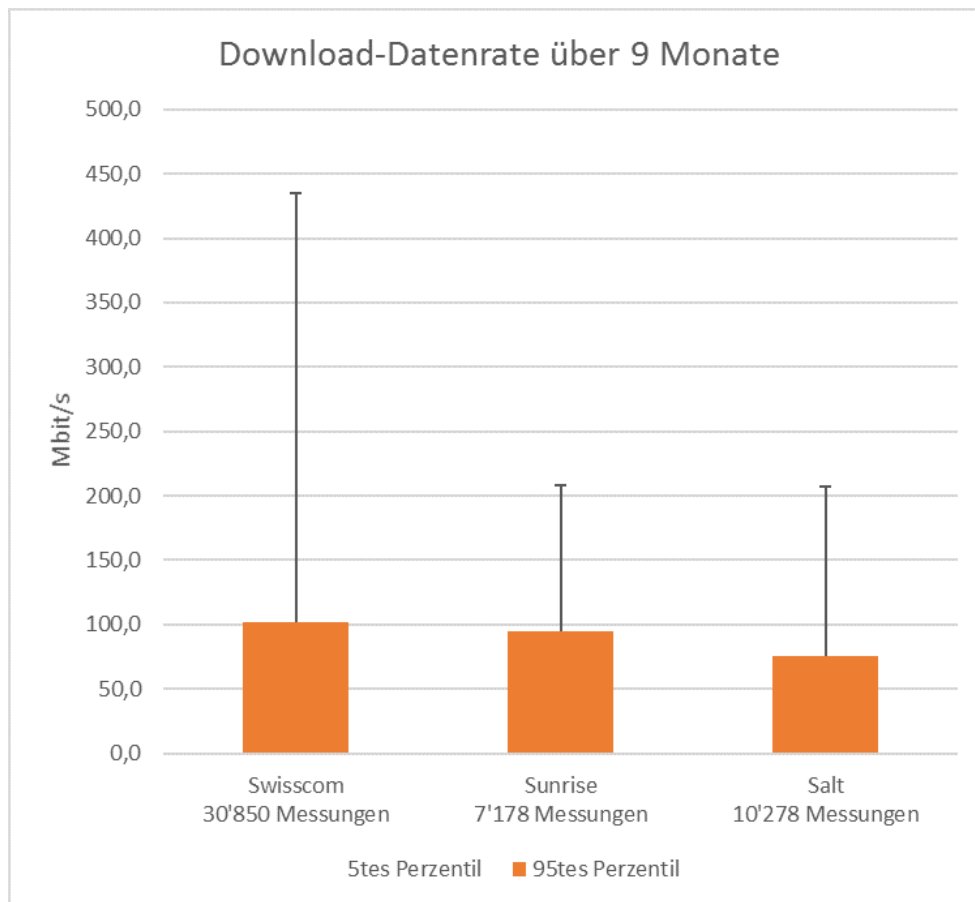


Abbildung 3.6: Darstellung der problematischen Ausreisser, welche die höchsten 5% der Messungen ausmachen

## 3.2 Zufallsexperiment

Es stellt sich nun die Frage, ab wann ein 50tes Perzentil, wie im Abschnitt 3.1.1 berechnet, „zuverlässig“ ist. Um dies zu beantworten, wurde pro Provider die Download-Datenrate bei 10 Durchläufen mit  $N$  zufällig gewählten Messwerten berechnet und ausgewertet. Zuverlässig bedeutet, dass der Durchschnitt des 50ten Perzentils über diese 10 Messungen nicht mehr als  $\pm 5\%$  vom 50ten Perzentil aller Messungen des jeweiligen Providers abweicht. Die Ergebnisse zu den einzelnen Providern sowie die Betrachtung über alle Provider zusammen werden in den folgenden Abschnitten erläutert.

### Betrachtung von Swisscom

Bei Swisscom liegen 30'850 Messungen vor. Ab einer zufälligen Wahl von 4'000 Messungen kann man sich auf die Aussage des 50ten Perzentils verlassen, da ab dann die durchschnittlichen Abweichungen vom 50ten Perzentil aller Swisscom-Messungen bei unter  $\pm 5\%$  liegen. Dies ist auch in der Abbildung 3.7 ersichtlich.

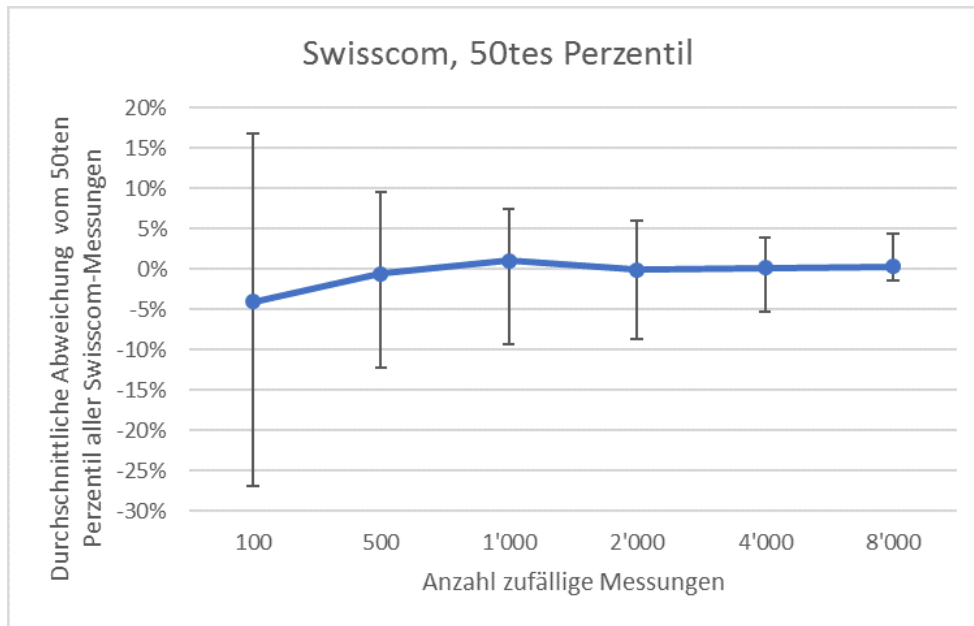


Abbildung 3.7: Darstellung des Verlaufs der Abweichung um das 50te Perzentil bei zunehmender Anzahl von zufällig gewählten Messungen

Damit sich die resultierenden Abweichungen vergleichen lassen, wurden nicht die gleiche Anzahl Messungen wie bei Swisscom, sondern der gleiche Anteil an der Anzahl totalen Messungen pro Provider ausgewertet. Der daraus entstandene Vergleich der Abweichungen pro Provider ist im letzten Abschnitt dieses Kapitels aufgeführt.

### Betrachtung von Sunrise

7'178 Messungen liegen bei Sunrise vor. Ab einer zufälligen Wahl von ungefähr 2'000 Messungen kann man sich auf die Aussage des 50ten Perzentils verlassen, da ab dann die durchschnittlichen Abweichungen bei unter  $\pm 5\%$  liegen, was in Abbildung 3.8 ersichtlich ist.

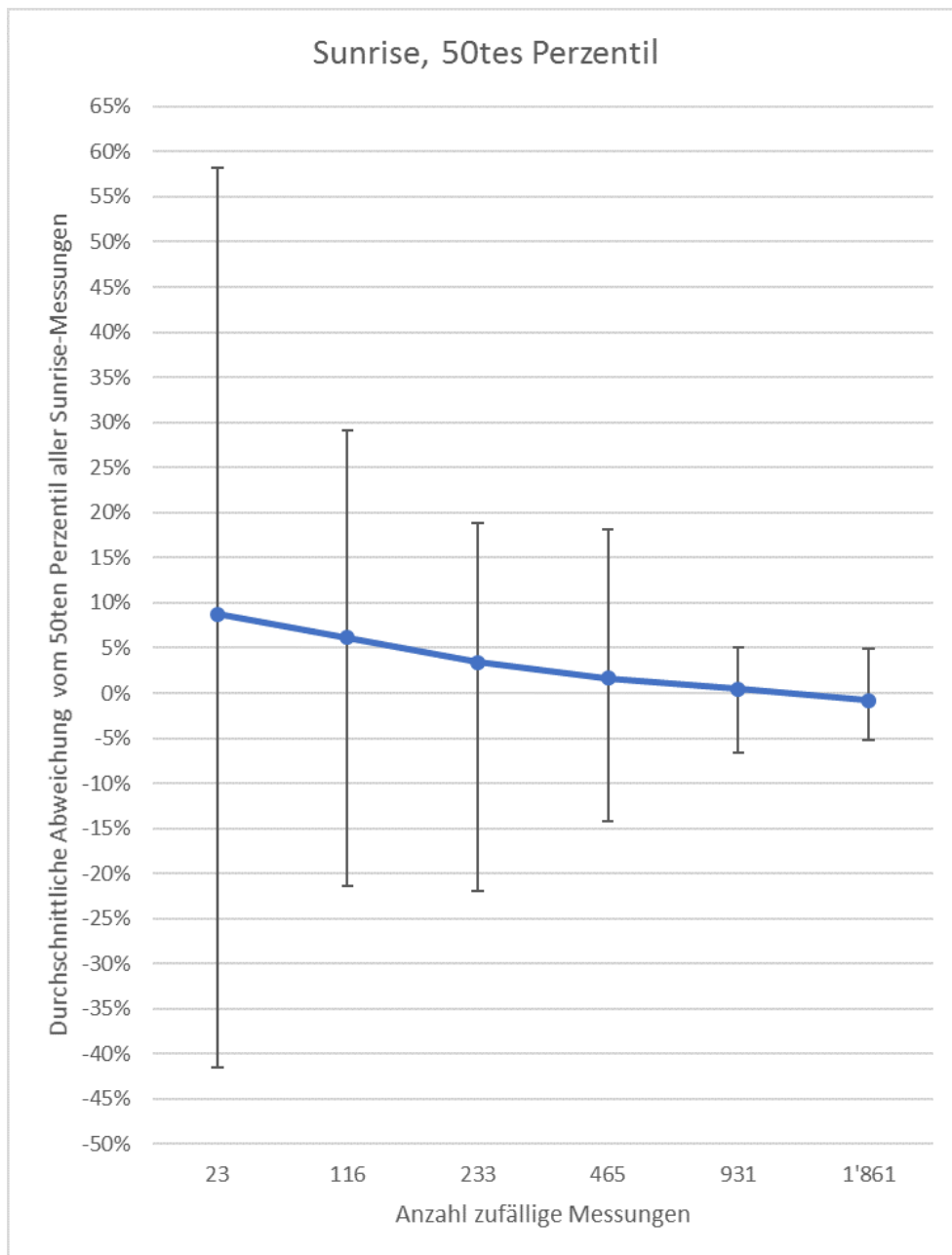


Abbildung 3.8: Darstellung des Verlaufs der Abweichung um das 50te Perzentil bei zunehmender Anzahl von zufällig gewählten Messungen

### Betrachtung von Salt

Salt hat total 10'278 Messungen. Ab ca. 3'600 zufällig gewählten Messungen liegen die durchschnittlichen Abweichungen unter der Grenze von +/-5%. Die Abbildung 3.9 zeigt dies auf.

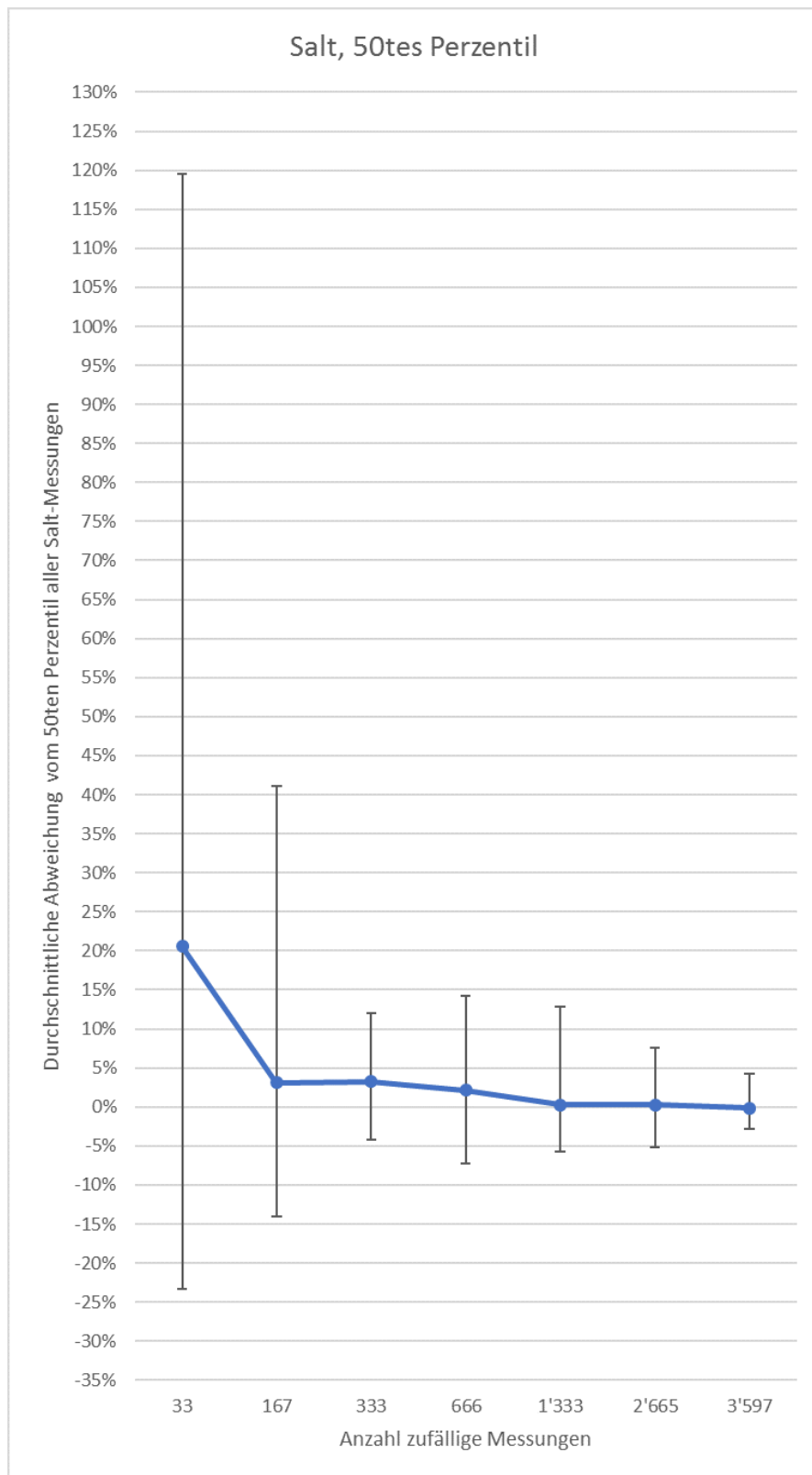


Abbildung 3.9: Darstellung des Verlaufs der Abweichung um das 50te Perzentil bei zunehmender Anzahl von zufällig gewählten Messungen  
41 von 142

**Betrachtung aller Providern zusammen**

Über alle Provider hinweg gibt es Total 48'306 Messungen. Hier kann man sich ab einer zufälligen Wahl von ungefähr 6'500 Messungen auf die Aussage des 50ten Perzentils verlassen, was aus der Abbildung 3.10 ersichtlich ist.

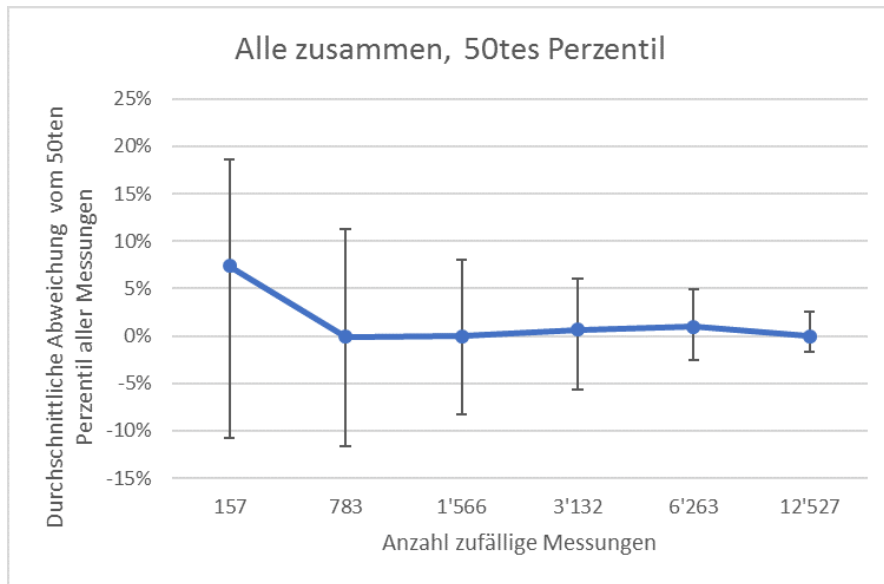


Abbildung 3.10: Darstellung des Verlaufs der Abweichung um das 50te Perzentil bei zunehmender Anzahl von zufällig gewählten Messungen

In der folgenden Darstellung werden die Verläufe der durchschnittlichen Abweichungen des 50ten Perzentils aller oben ausgewerteten Provider sowie aller Provider zusammen aufgezeigt:

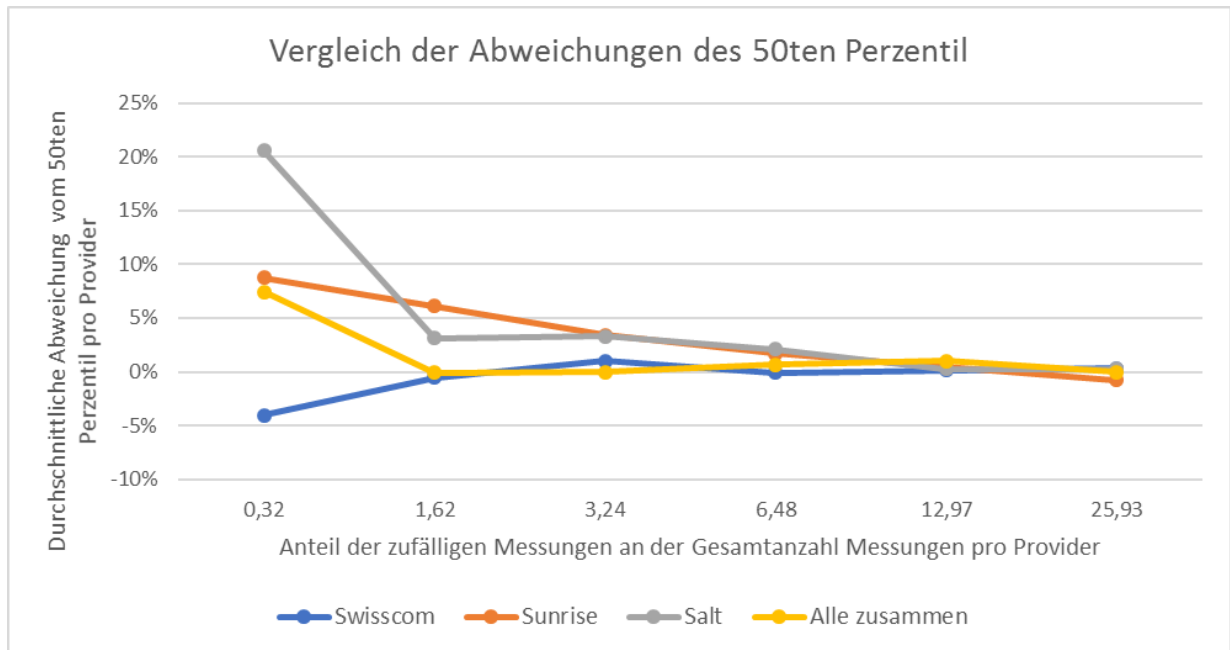


Abbildung 3.11: Zusammenführung aller Verläufe um das 50te Perzentil bei zunehmender Anzahl von zufällig gewählten Messungen

### Abweichungen der Minimal- und Maximalwerte bei N Messungen

Betrachtet man sich bei den N Messungen die Abweichungen, welche die N Minimalwerte bzw. Maximalwerte zum entsprechenden Wert aller Messungen aufweisen, so stellt man fest, dass diese sehr gross ausfallen.

Bei Swisscom liegt die grösste Abweichung des Minimum zum Minimum aller Swisscom-Messungen auch bei 8'000 zufällig gewählten Messungen noch bei 12'600%. Der Maximalwert weicht bei dieser Anzahl Messungen bis zu -14% ab.

Grosse Abweichungen sind auch bei Sunrise (1'861 Messungen, Minimum: Abweichungen bis zu 882%, Maximum: Abweichungen bis zu -18%) und bei Salt (3'597 Messungen, Minimum: Abweichungen bis zu 13'064%, Maximum: Abweichungen bis zu -10%) ersichtlich. Das gleiche Bild zeigt sich auch, wenn man die zufällige Auswahl bei allen Providern zusammen durchführt (12'527 Messungen, Minimum: Abweichungen bis zu 10'833%, Maximum: Abweichungen bis zu -15%).

Es wird generell empfohlen, keine Aussagen zu minimalen oder maximalen Werten aufgrund einer zufälligen Auswahl von Messwerten zu machen.

### 3.3 Monatliche Auswertung

Um zu überprüfen, wie stark die Abweichungen sind, wenn jeder Monat für sich betrachtet wird, wurde für jeden Provider das 50te Perzentil der Download-Datenrate pro Monat ausgewertet. Anschliessend wurde das monatliche Ergebnis mit dem 50ten Perzentil aller Messungen des jeweiligen Providers verglichen.

Wie anhand der Abbildungen von Swisscom 3.12, Sunrise 3.13 und Salt 3.14 ersichtlich ist, gibt es starke Abweichungen der einzelnen Monate. Bei Swisscom beträgt die Abweichung des 50ten Perzentils im Monat März 21,83%, bei Sunrise liegt die grösste Abweichung bei -40,53% im Monat Dezember und bei Salt weicht der Monat Juli um 36,73% vom 50ten Perzentil über alle Messungen der Salt ab.

Es wird deshalb empfohlen, bei einer Auswertung stets darauf zu achten, dass sich Daten vom ganzen Auswertungszeitraum darin befinden.

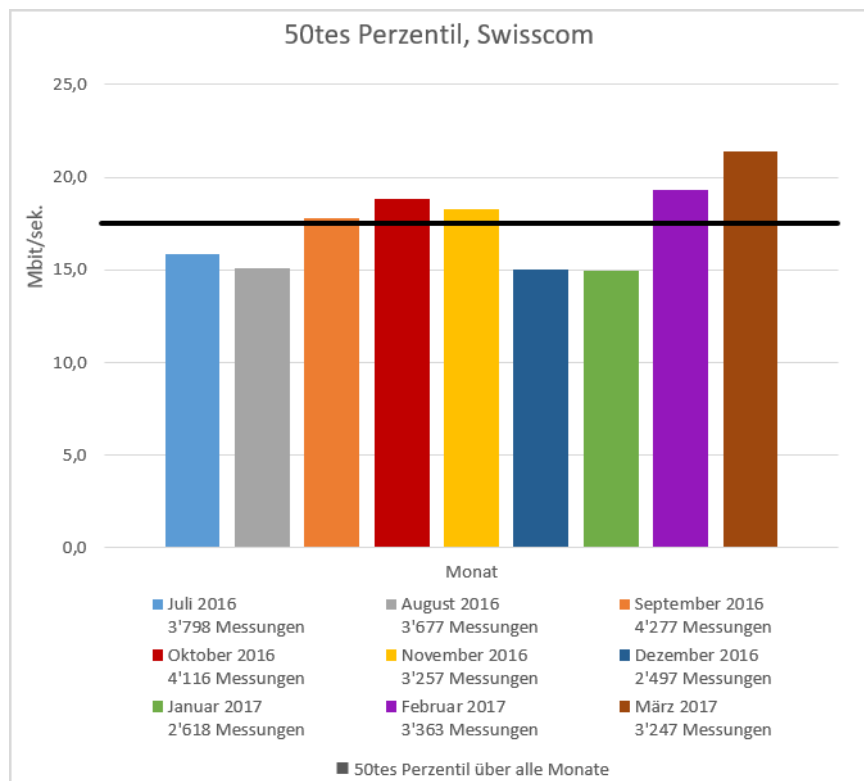


Abbildung 3.12: Monatliche Auswertung des 50ten Perzentils bei Swisscom

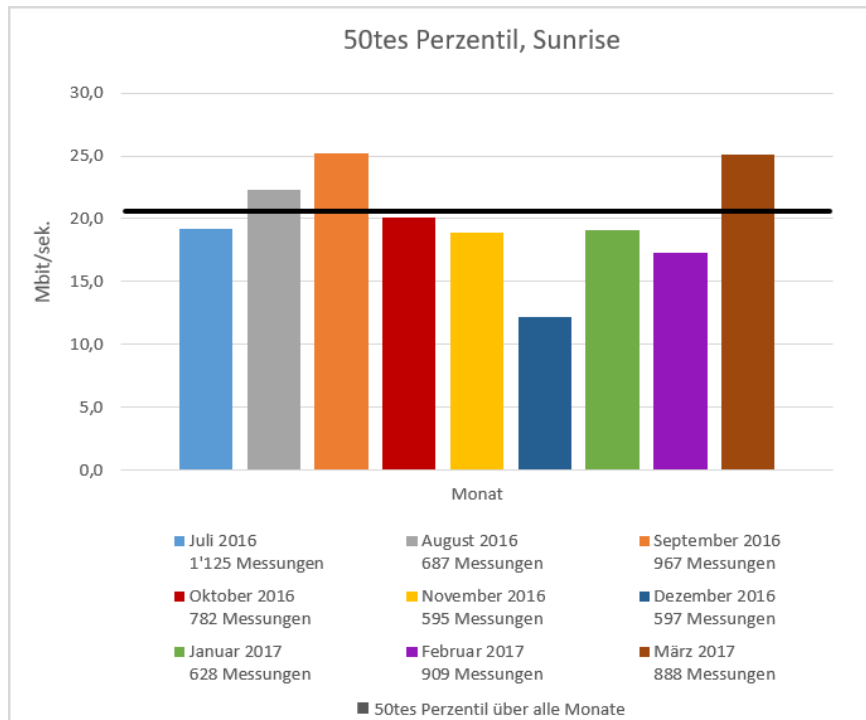


Abbildung 3.13: Monatliche Auswertung des 50ten Perzentils bei Sunrise

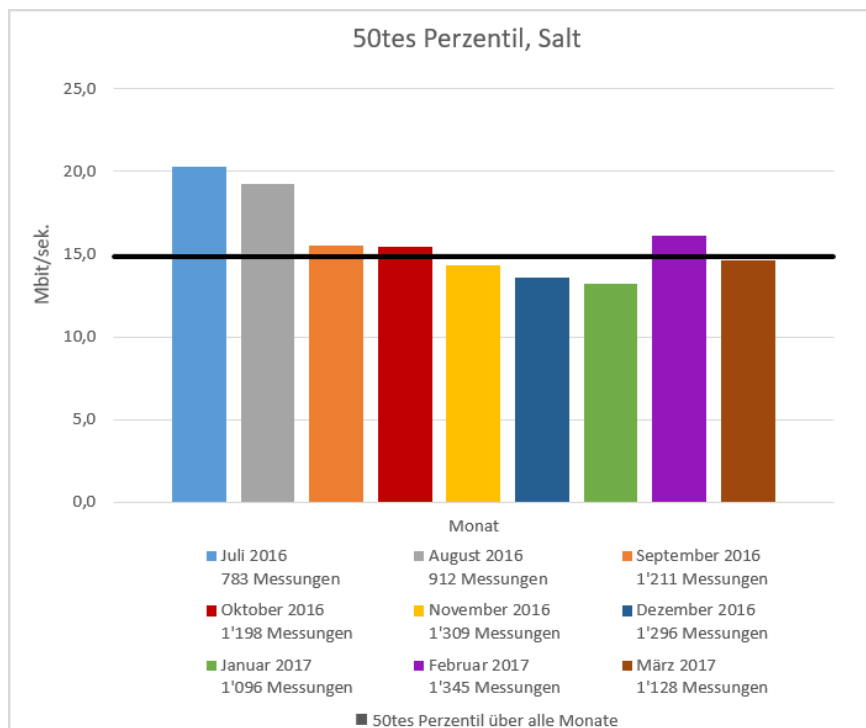


Abbildung 3.14: Monatliche Auswertung des 50ten Perzentils bei Salt

### 3.4 Regionale Auswertung

Im Kapitel 2.1 „Aussagekraft von Netztests“ wurde die Aussage aufgestellt, dass Messergebnisse in städtischen Gebieten anders ausfallen, als auf dem Land. Dies ist einer der Punkte, welche bei Drive- und Walk-Tests beachtet werden sollte. Ziel dieser regionalen Auswertung war es, die erwähnte Aussage mit Speedtest-Resultaten der cnlab AG zu untermauern.

In der Abbildung 3.15 ist die Gegenüberstellung von Download-Datenraten der grössten Stadt zum Rest des Kantones (exklusive der grössten Stadt) ersichtlich. Dabei wurden die Kantone Zürich und St. Gallen verwendet. Es ist erkennbar, dass es tatsächlich einen beträchtlichen Unterschied zwischen der Stadt und dem Rest des Kantones gibt. Ausgehend von der Stadt als 100% beträgt die Abweichung beim Kanton Zürich -17,26% und beim Kanton St. Gallen -51,46%. Es wird deshalb empfohlen, dass bei der Auswertung von Speedtest-Messungen sowohl Resultate von städtischen, als auch von ländlichen Regionen in das Ergebnis miteinfließen. Bei der Betrachtung von Netztests ist unbedingt die regionale Verteilung der Messungen anzuschauen. Falls sich ein Test zu stark auf die grossen Städte konzentriert, zeichnet sich ein falsches Bild mit zu hohen Download-Datenraten ab.

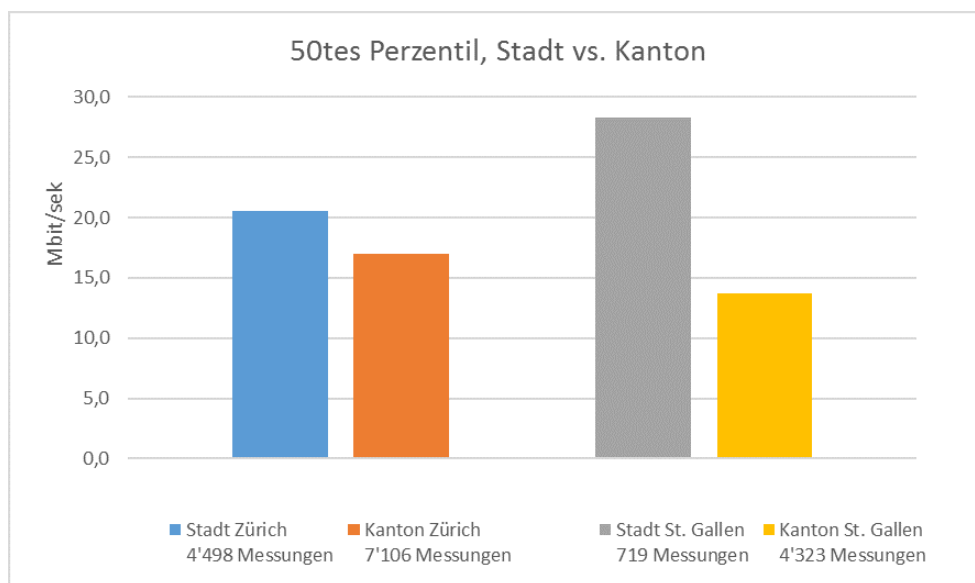


Abbildung 3.15: Vergleich zwischen Stadt und Kanton

### 3.5 Schlussfolgerungen

Die wichtigsten Erkenntnisse aus den statistischen Auswertungen sind:

- Aufgrund einer zufälligen Auswahl von Messwerten sollten keine Aussagen zu minimalen oder maximalen Werten gemacht werden.

- Um bei einer zufälligen Auswahl von Werten eine zuverlässige Aussage zum 50ten Perzentil machen zu können, benötigt es bei Swisscom ca. 4'000 Werte (12,92%), bei Sunrise ca. 2'000 gewählte Messungen (25,93%), bei Salt ca. 3'600 Werte (35,00%) und bei allen Providern zusammen ca. 6'500 gewählte Werte (12,92%).
- In eine statistische Auswertung sollte der ganze Auswertungszeitraum miteinbezogen werden, da es zwischen den einzelnen Monaten zu beträchtlichen Schwankungen kommen kann.
- Bei der regionalen Verteilung von Messwerten soll darauf geachtet werden, dass die städtischen Werte zu keiner Verzerrung führen, dass also auch Werte aus den ländlichen Regionen miteinfließen.

# Kapitel 4

## Verwendete Datenquellen

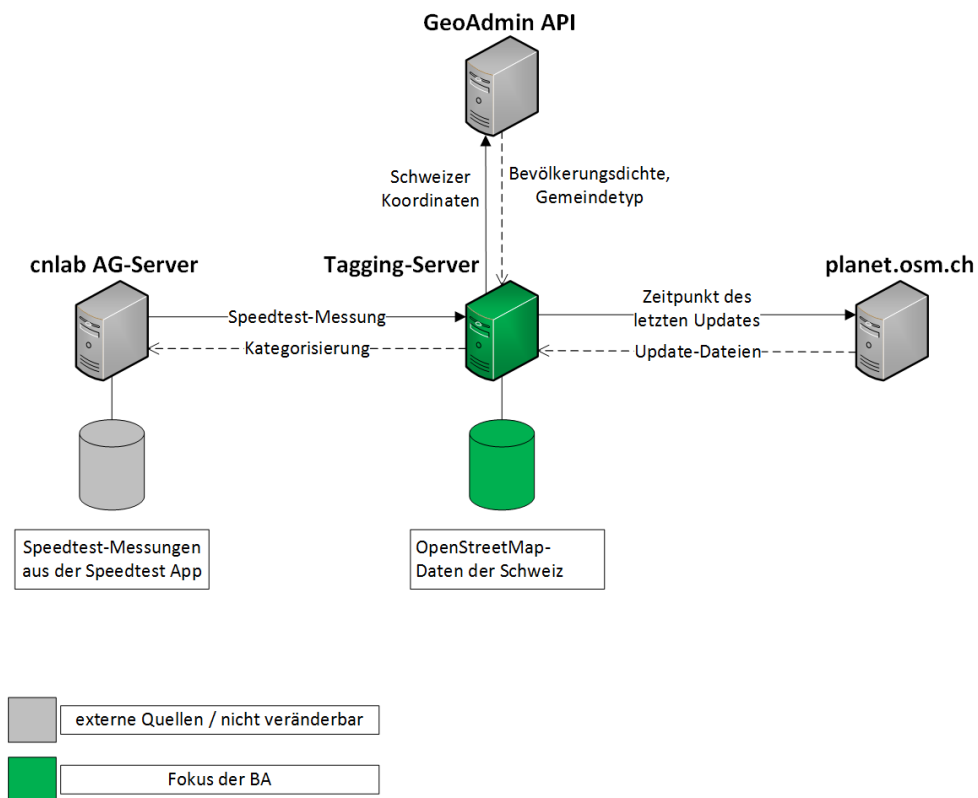


Abbildung 4.1: Übersicht der Datenquellen

In der Abbildung 4.1 ist der in dieser Arbeit entwickelte Tagging-Server in grün dargestellt. Stellt ein Server, in diesem Beispiel jener der cnlab AG, eine Tagging-Anfrage, so werden Daten von zwei externen Datenquellen benötigt. Die erste Quelle ist das GeoAdmin API des Bundes [1], welches die Bevölkerungsdichte sowie den Gemeindetyp für die Messung zurückliefert. Bei der zweiten Quelle handelt es sich um OpenStreetMap-Daten, welche aus der lokalen Datenbank

geladen werden. Die Datenbank wird stündlich mittels Update-Dateien von planet.osm.ch aktualisiert.

In den folgenden Abschnitten wird auf die Lizenzen der einzelnen Datenquellen genauer eingegangen.

## **4.1 Lizenzabklärungen OpenStreetMap (OSM)**

OpenStreetMap ist ein Projekt mit dem Ziel, eine frei zugängliche Weltkarte zu erschaffen [2]. Die erfassten Daten der Schweiz werden genutzt, um für eine bestimmte Speedtest-Messung den Ort und die geografische Umgebung herausfinden zu können. Wie die Daten genau genutzt werden, ist im Kapitel 6.4.2 „Bestimmung der location“ beschrieben.

Gemäss Abklärungen mit Herr Keller dürfen die OpenStreetMap-Daten für den im Abschnitt 6.1 beschriebenen Use Case verwendet werden. Das entsprechende Mail ist im Anhang unter 14.8.1 „Lizenzabklärung OSM“ angefügt.

## **4.2 Lizenzabklärung GeoAdmin API**

Die angefragten Dienste des GeoAdmin API verwenden Daten des Bundesamtes für Raumentwicklung ARE [52]. Wie diese Daten genau verwendet werden, ist in den Kapiteln 6.4.3 und 6.4.4 detailliert aufgezeigt.

Damit diese Daten genutzt werden dürfen, muss gemäss dem Disclaimer der Bundesverwaltung [53] die schriftliche Zustimmung der Urheberrechtsträger eingeholt werden. Die Bewilligung für die Bachelorarbeit ist im Anhang unter Kapitel 14.8.2 aufgeführt.

Falls das Endprodukt der Bachelorarbeit durch die cnlab AG eingesetzt wird, muss die cnlab AG diese Bewilligung für sich ebenfalls einholen. Dazu muss der Datenherr des Datenbestands kontaktiert werden, welcher im geocat (siehe Abbildung 4.2) gefunden werden kann.

Gemeindetypologie ARE

Überblick Verbunden **Vollständig** Basisinfo Vertriebsinfo Metadaten

**Basisinformation zum Datenbestand**

*Kurzbeschreibung* Die Gemeindetypen ARE sind das Ergebnis einer Kombination zwischen den Grossregionen, der Agglomerationsdefinition 2000 sowie der Gemeindetypologie des Bundesamts für Statistik BFS. Aus den ursprünglich 13 Typen wurde eine Typologie bestehend aus 9 Typen abgeleitet.

*Sprache* Deutsch

*Zeichensatz* Utf8

**Thematik**

E1 Raumplanung, Raumentwicklung

**Bibliografische Angaben**

*Titel* Gemeindetypologie ARE

*Alternativtitel* Gemeindetypologie ARE

*Erstellung* 01-01-2014

**Kontakt für die Ressource**

**Allgemein**

*Person* Rolf Giezendanner

*Organisation* Bundesamt für Raumentwicklung

*Abkürzung der Organisation* ARE

*Funktion* Eigentümer / Datenherr

**Adressangaben**

*Ort* Ittigen

*PLZ* 3063

*Staat* CH

*E-Mail* rolf.giezendanner@are.admin.ch

*Strassenname* Worblentalstrasse

*Hausnummer* 66

**Telefon**

*Direkte Nummer* +41 58 462 01 43

**Online-Ressource**

*URL* <http://www.aren.admin.ch>

*Protokoll* text/html

Abbildung 4.2: Auszug aus den geocat Informationen [54]

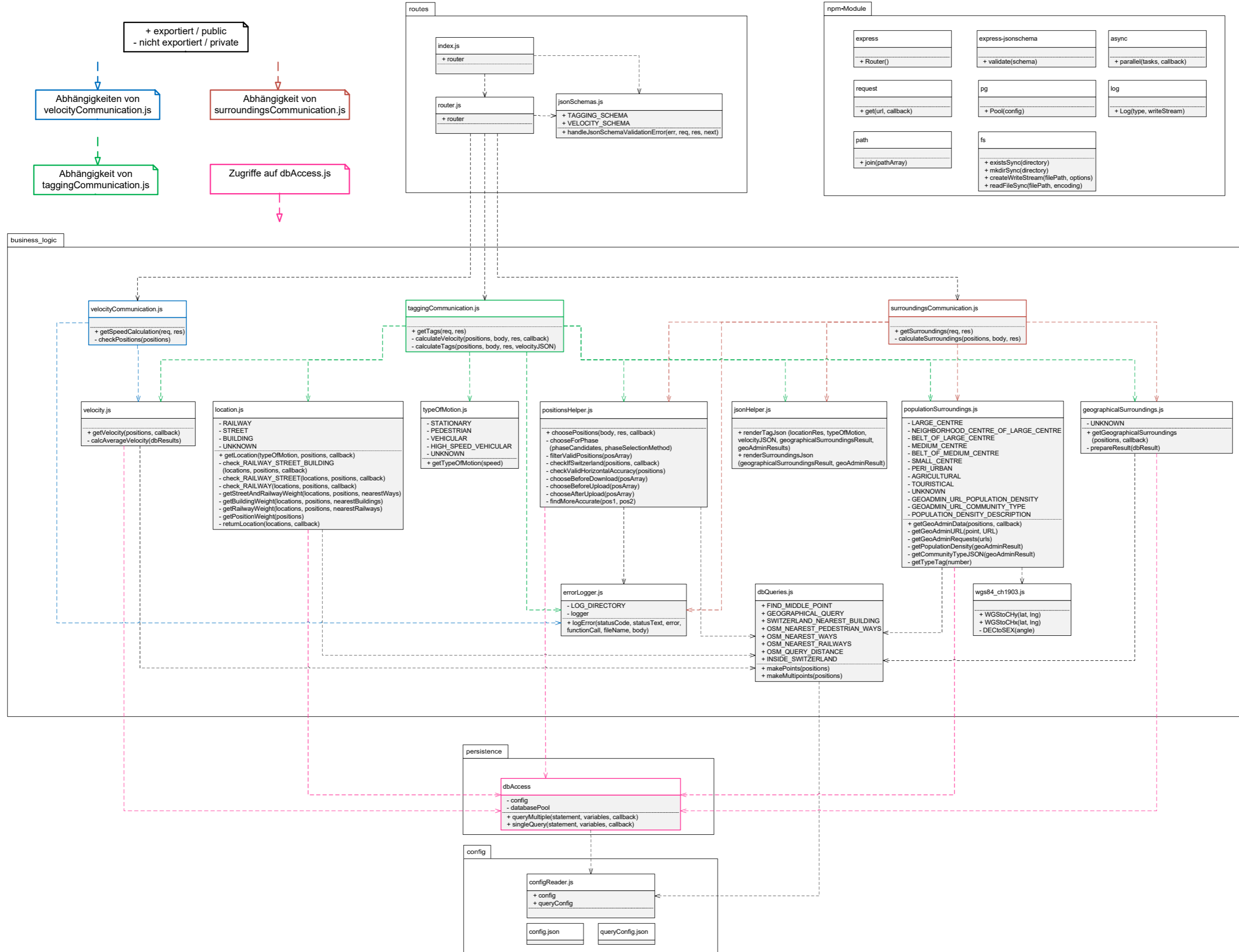
Bei der Anfrage für die Bewilligung der Datennutzung muss der Einsatzzweck und die Organisation, welche die Daten nutzt, beschrieben werden.

## **Kapitel 5**

# **Software Engineering**

Auf der folgenden Seite ist das Datei-Diagramm des Tagging-Servers abgebildet. Dieses zeigt deren Abhängigkeiten untereinander, sowie deren Einteilung in einzelne Software-Layer auf. In diesem Kapitel werden die Aufgaben der Layer wie auch der einzelnen Dateien erläutert. Es dient dazu, sich einen schnellen Überblick über den gesamten Code machen zu können.

# Datei-Diagramm Tagging-Server



## 5.1 Routes (Layer 1)

Der erste Layer dient der Prüfung und Weiterleitung von Input-Daten.

### **index.js**

In dieser Datei werden die Versionsunabhängigen Anfragen behandelt. Dazu gehört z.B. das Rendering der API-Seite. Versionsspezifische Anfragen werden an router.js weitergereicht.

### **router.js**

Für jede Version wurde ein eigener Router erstellt. Dieser ist für die Weiterleitung von Tagging-, Velocity- und Surroundings-Abfragen an die jeweilige „Communications“-Datei zuständig. Bevor die Input-Daten allerdings nach unten weitergereicht werden, findet eine Validation des Inputs statt.

### **jsonSchemas.js**

Diese Datei beinhaltet die definierten Schemas zur Validation der Input-Daten, wie sie im Abschnitt 6.2.3 genauer erläutert wird. Zudem werden darin Input-Validations-Fehler beantwortet.

## 5.2 Business logic (Layer 2)

Im zweiten Layer befindet sich die gesamte Logik, um das Tagging zu erstellen. Von oben werden Anfragen mit korrekt formatierten Input-Daten entgegengenommen und beantwortet. Nach unten werden Queries an die Datenbank geschickt, um deren Resultat auszuwerten.

Für die Entgegennahme der Anfrage gibt es für jede API-Route (/tag, /calculateVelocity und /getSurroundings) eine eigene „Communication“-Datei, welche die Koordination zwischen den Logik-Berechnungen in den darunterliegenden Dateien übernimmt.

Alle Dateien, welche auf die Datenbank zugreifen, haben ebenfalls eine Abhängigkeit auf die Datei „dbQueries.js“, da sie von dort die benötigten Datenbankabfragen holen.

Die Reihenfolge der nachfolgenden Dateien wurde so gewählt, dass die Informationen aufeinander aufbauen. Zuerst wird die überschaubare Implementation der Geschwindigkeit betrachtet, dann die Berechnung der Umgebung, um schliesslich auf das Tagging einzugehen. Dieses benutzt die bis dahin besprochenen Dateien, plus jene für die Bestimmung des Messorts.

### 5.2.1 Velocity

#### **velocityCommunication.js**

Die Datei prüft die Input-Daten auf ihre Validität und leitet Anfragen an die Geschwindigkeitsberechnung in „velocity.js“ weiter.

#### **velocity.js**

Für alle eingehenden Positionen wird hier die Distanz, Zeit sowie die Durchschnittsgeschwindigkeit berechnet. Für die Bestimmung der Distanz wird die Datenbank abgefragt.

Auf die dahinterstehende Logik wird im Abschnitt 6.4.1 genauer eingegangen.

## 5.2.2 Surroundings

### **surroundingsCommunication.js**

Diese Datei wählt mit „positionsHelper.js“ die besten Input-Daten aus, reicht diese an „geographicalSurroundings“ und „populationSurroundings“ weiter und bereitet die Resultate mit „jsonHelper.js“ für die Rückgabe auf.

### **geographicalSurroundings**

Die Bestimmung der geografischen Umgebung um den Messpunkt wird in dieser Datei berechnet. Dafür wird die Datenbank nach bestimmten Keys durchsucht. Das genaue Vorgehen ist in Abschnitt 6.4.5 erläutert.

### **populationSurroundings**

In der Datei „populationSurroundings.js“ wird die Bestimmung von community type (siehe Abschnitt 6.4.3) und population density (siehe Abschnitt 6.4.4) vorgenommen. Dazu wird in beiden Fällen das GeoAdmin API angefragt. Für die Koordinaten-Umwandlungen wird die Datei „wgs84\_ch1903.js“ verwendet.

### **wgs84\_ch1903.js**

Da das GeoAdmin API [1] die Input-Koordinaten im CH1903-Format [55] [56] verlangt, müssen die im WGS 84-Format vorliegenden Koordinaten der Speedtest-Messung umgewandelt werden. Dies wird mit dieser Datei erledigt, welche von swisstopo unter folgender URL zur Verfügung gestellt wird:

<https://cms.geo.admin.ch/www.swisstopo.admin.ch/archives/cms2007/internet/swisstopo/de/home/products/software/products/skripts.html>

## 5.2.3 Tagging

### **taggingCommunication.js**

Diese Datei koordiniert die Tagging-Anfragen. Dazu wird zuerst mittels „velocity.js“ die Geschwindigkeit berechnet, um anschliessend mit Hilfe von „typeOfMotion.js“ den Fortbewegungstyp zu bestimmen. Diese Daten fliessen in die Berechnung von location mit ein, welche parallel zur Berechnung vom community type, population density und geographical surrounding läuft. Schlussendlich werden die Ergebnisse mit Hilfe des „jsonHelper.js“ zur Antwort an den Client zusammengesetzt.

### **positionsHelper.js**

Hier werden die korrekt formatierten Input-Daten auf ihre Validität geprüft. Um herauszufinden, ob alle erhaltenen Punkte in der Schweiz liegen, wird die Datenbank angefragt. Welche weiteren Prüfungen vorgenommen werden, wird in Abschnitt 6.2.3 genauer erklärt. Anschliessend werden die genauesten Daten für die Weiterverarbeitung ausgewählt. Auf diesen Prozess wird in Kapitel 6.2.4 eingegangen.

### **typeOfMotion.js**

Anhand der in „velocity.js“ berechneten Geschwindigkeit wird der type of motion bestimmt. Der Abschnitt 6.4.1 befasst sich mit diesem Vorgang.

### **location.js**

In dieser Datei wird die Berechnung der location durchgeführt. Dazu wird einerseits der type of motion verwendet und andererseits die Datenbank nach Schienen, Strassen oder Gebäuden in der Umgebung abgefragt. Wie daraus die location bestimmt wird, ist in Abschnitt 6.4.2 beschrieben.

### **jsonHelper.js**

Diese Datei wird benötigt, um die Resultate aus „taggingCommunication.js“ und „surroundingsCommunication.js“ als einheitliche JSON-Antworten an den Client zurückzusenden. Wie diese genau Aussehen, kann im Abschnitt 6.3 nachgelesen werden.

## **5.2.4 Weitere Dateien**

### **dbQueries.js**

Diese Datei beinhaltet alle Datenbank-Queries zentral an einem Ort. Zudem bietet sie Hilfsfunktionen an, um die Queries dynamisch zu erstellen.

### **errorLogger.js**

Fehler, welche während der Laufzeit auftreten, werden an verschiedenen Stellen im Code abgefangen und eine Statusmeldung 500 „Internal Server Error“ an den Client zurückgeschickt. Um solche Fehlerfälle anschliessend nachvollziehen und damit auch beheben zu können, werden diese in der Datei „errorLogger.js“ in ein Log-File geschrieben.

## **5.3 Persistence (Layer 3)**

Der dritte Layer ist für den Zugriff auf die Datenbank zuständig.

### **dbAccess.js**

In dieser Datei wird auf die Open Street Map (OSM)-Datenbank zugegriffen. Dabei werden der Datenbankname, das Passwort sowie weitere Werte, mit Hilfe von „configReader.js“ aus der Datei „config.json“ gelesen. Anschliessend wird damit ein Pool aufgesetzt, bei welchem eine bestimmte Anzahl von Datenbank-Clients ständig verbunden sind. Im Fall des Tagging-Servers sind es 10 Clients. Auf diese Weise können pro Abfrage 20-30ms eingespart werden, da der Verbindungsaufbau wegfällt. Zudem kann pro Client nur 1 Abfrage gleichzeitig abgearbeitet werden, wodurch mehrere verbundene Clients einen klaren Zeitvorteil mit sich bringen. [57]

Nach diesem Aufbau werden Funktionen für eine einzelne sowie für mehrere, parallel ablaufende Datenbank-Abfragen angeboten.

## 5.4 Config (Layer 4)

Der vierte Layer dient dazu, den Tagging-Server flexibel einsetzen zu können, indem z.B. die Parameter für die Datenbankverbindung in Konfigurationsdateien ausgelagert werden. Somit ist für das Ändern dieser Werte kein Eingriff im Code nötig.

### **configReader.js**

Diese Datei dient dem Auslesen der Konfigurationsdateien.

### **config.json**

Darin ist der Server-Port sowie die Verbindungsparameter für die Datenbank hinterlegt.

### **queryConfig.json**

Die Datei hilft dabei, die Query-Parameter der Datenbankabfragen individuell anzupassen. So wird darin z.B. die Distanz spezifiziert, in welcher das nächste Gebäude gesucht werden soll.

## 5.5 npm-Module

In diese Kategorie fallen alle Libraries, welche vom Node Package Manager (npm) bezogen wurden. Diese werden an verschiedenen Stellen im Code benötigt, wurden aber aus Gründen der Übersichtlichkeit separat abgebildet. Zudem ist eine zyklische Abhängigkeit von Dateien ausgeschlossen, da diese Module keine der Tagging-Server-Klassen benötigt.

Was npm genau ist und welche Module wozu verwendet werden, ist im Abschnitt 14.2 beschrieben.

## Kapitel 6

# Realisierung

Das Ziel des Tagging-Servers ist es, bestehende Speedtest-Messdaten aus der cnlab App [17] zu klassieren. Die cnlab AG soll damit verbesserte statistische Aussagen zu den Speedtest-Ergebnissen ihrer App-Benutzer machen können. Dies ist möglich, in dem zum Beispiel untersucht wird, welche Unterschiede es zwischen stationären und Hochgeschwindigkeits-Messungen gibt. Weiter können Differenzen zwischen Messungen auf Schienen und Strassen genauer betrachtet werden.

In diesem Kapitel werden die wichtigsten Punkte der Implementation des Tagging-Servers, welcher mit Node.js umgesetzt wurde, genauer erläutert. Die Kriterien, anhand derer eine Klassierung vorgenommen werden soll, sind im Abschnitt „Use Case“ 6.1 definiert.

Zum Zeitpunkt der Abgabe kann der Algorithmus ausschliesslich mit Mess-Daten aus der Schweiz umgehen. Welche weiteren Einschränkungen es gibt, welche Daten zur Klassierung verwendet werden und welches Format diese aufweisen müssen, wird im Unterkapitel „Input-Daten“ 6.2 erläutert. Anschliessend wird aufgezeigt wie die Antworten des Servers aussehen (siehe Abschnitt „Output-Daten“ 6.3) und wie diese berechnet werden (siehe Unterkapitel „Logik“ 6.4).

Eine lauffähige Instanz des Tagging-Servers ist unter folgender URL erreichbar:  
<http://sinv-56041.edu.hsr.ch/api/v5.1/tag>

Neben der Pfad „/tag“, welcher die Kategorisierung vornimmt, stehen auch „/calculateSpeed“ für die Geschwindigkeitsberechnung und „/findSurroundings“ für die Abfrage der Umgebung zur Verfügung. Mit diesen beiden Pfaden können Teilberechnungen des Taggings separat abgefragt werden. Wie die Anfragen an den Server zu stellen sind, kann in der API-Beschreibung unter folgender URL nachgelesen werden:

<http://sinv-56041.edu.hsr.ch/api>

## 6.1 Use Case: Mapping Geodaten auf Bewegungseigenschaften

### 6.1.1 Ursprung der Daten

Bei den Speedtest-Messungen der cnlab Speed Test App gibt es die folgenden vier Phasen.

- First Connection Time (FCT)  
Zeit bis zur ersten Verbindungsbestätigung des Referenz-Servers
- Download  
Zeit, in welcher die Download-Geschwindigkeit gemessen wird
- Upload  
Zeit, in welcher die Upload-Geschwindigkeit gemessen wird
- Round Trip Time (RTT)  
Zeit, um ein Datenpaket zum Referenz-Server und zurück zu übertragen

Am Anfang und Ende jeder Phase werden die Standort-Daten sowie die Zeit der Testperson erfasst. Die Speicherung dieser Daten bei der cnlab AG ist in der folgenden Tabelle, anhand einer Beispiel-Messung, abgebildet. Es werden intern noch weitere Daten erfasst, welche allerdings für diese Arbeit nicht von Belang sind, deshalb werden diese Daten nicht aufgeführt. Die Erklärungen der einzelnen Spalten sind in den Abschnitten 6.2.1 und 6.2.2 vorzufinden.

MeasurementID	ID	Date	Phase	Latitude	Longitude	Altitude	Horizontal Accuracy	Vertical Accuracy	GPS Lock
41141837	1	2017-03-28 07:31:44.0	FCTStart	47.3589998	8.7095882	0	800	-1	1
41141837	2	2017-03-28 07:31:44.0	FCTEnd	47.3589998	8.7095882	0	800	-1	1
41141837	3	2017-03-28 07:31:44.0	DownloadStart	47.3589998	8.7095882	0	800	-1	1
41141837	4	2017-03-28 07:31:54.0	DownloadEnd	47.3530638	8.7135701	0	98.40000 15258789	-1	1
41141837	5	2017-03-28 07:31:54.0	UploadStart	47.3530638	8.7135701	0	98.40000 15258789	-1	1
41141837	6	2017-03-28 07:32:06.0	UploadEnd	47.3516764	8.7165203	0	82.5	-1	1
41141837	7	2017-03-28 07:32:06.0	RTTStart	47.3516764	8.7165203	0	82.5	-1	1
41141837	8	2017-03-28 07:32:07.0	RTTEnd	47.3516764	8.7165203	0	82.5	-1	1

Abbildung 6.1: Mögliche Input-Daten der cnlab AG

Auf dem folgenden Bild werden die möglichen Input-Daten aus der Abbildung 6.1 mit ihren horizontalAccuracy-Werten dargestellt.

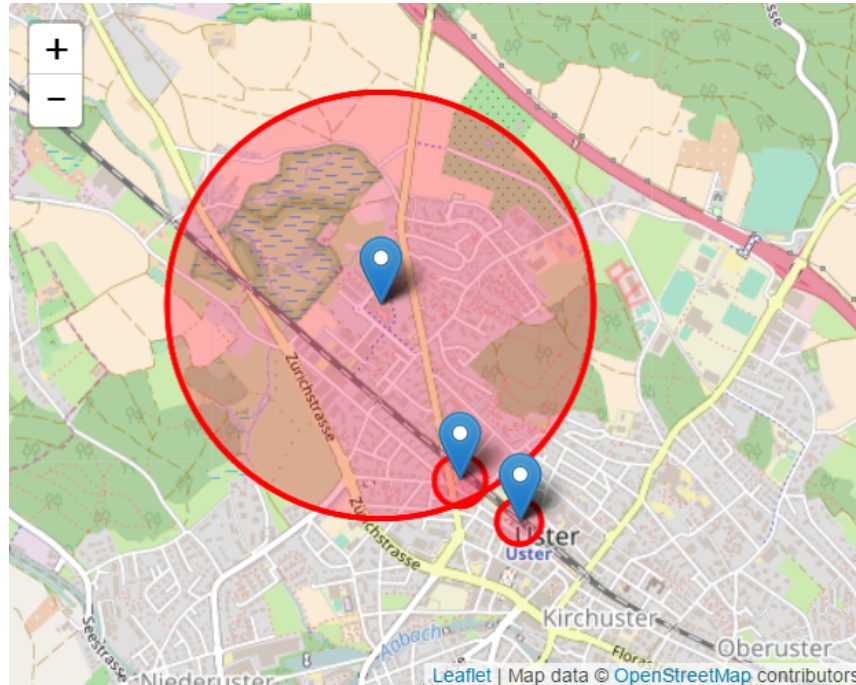


Abbildung 6.2: Darstellung der möglichen Input-Daten aus Abbildung 6.1 auf einer Leaflet-Karte [58]

### 6.1.2 Rückgabewerte

Die Daten werden im JavaScript Object Notation (JSON)-Format an den Tagging-Server gesendet. Dieser schickt als Antwort die folgenden Werte zurück:

1. Location (Messort)

In Klammer sind jeweils detailliertere Klassierungen angegeben.

- (a) Railway (Schiene: Fernverkehr, Nahverkehr, Tram, Bahnhof)
- (b) Street (Strasse: Hauptstrassen, Nebenstrassen, Autobahn)
- (c) Water (Wasser: Seen)
- (d) Building (Gebäude)
- (e) Forest (Wald)
- (f) Grassland (Wiese)
- (g) Unknown (Unbekannt)

2. Type of motion (Bewegungsart)

- (a) Stationary (Stationär: kleiner als 3 km/h)
- (b) Pedestrian (Fussgänger: zwischen 3 und 10 km/h)
- (c) Vehicular (Fahrzeuge: zwischen 10 und 140 km/h)
- (d) High speed vehicular (Hochgeschwindigkeitsfahrzeuge: zwischen 140 und 450km/h)

3. Population density (Besiedlungsdichte)

- (a) Rural (ländlich)
- (b) Urban (städtisch)

- Diese Daten werden ebenfalls im JSON-Format zurückgesendet. Dabei werden die englischen Bezeichnungen verwendet.
- Die location und der type of motion werden über die gesamte Messdauer ermittelt, die population density jeweils für Phase 2 (Download) und Phase 3 (Upload) separat.
- Die Angabe des type of motion ist dabei an die Klassierung gemäss „ETSI TR 101 115 V6.1.0 (2000-04)„[59] sowie an die Mobility Klassen aus dem Buch „LTE-Advanced“ [60] angelehnt.
- Züge fahren in der Schweiz mit Geschwindigkeiten bis 220 km/h [61]. Werden Speedtest-Messungen in Zügen durchgeführt, so resultieren oft ungenaue Positionen. Da dies zu einer höheren Geschwindigkeit als die eigentliche Fahrgeschwindigkeit führen kann, wurde die Grenze auf 450 km/h erhöht. Aus demselben Grund wurde Geschwindigkeit bei Fahrzeugen um 20 km/h höher gesetzt als die erlaubte Maximalgeschwindigkeit.

## 6.2 Input-Daten

### 6.2.1 Verwendete Daten

Für die Berechnung der Rückgaben werden von den acht Standorten pro Messung die folgenden Daten verwendet: (Im restlichen Teil der Dokumentation werden jeweils deren englischen Begriffe verwendet.)

- **Latitude (Breitengrad)**  
Der Breitengrad „beschreibt die Nord-Süd-Dimension eines Ortes auf der Erdoberfläche.“ In der Dezimalschreibweise, wie sie in der obigen Tabelle verwendet wird, liegen positive Werte nördlich und negative Werte südlich des Äquators. Mögliche Werte liegen zwischen -90 und +90 Grad und stellen den Winkel vom Äquator zum Messort dar. [62] Dieser Wert liegt im WGS 84-Format vor.
- **Longitude (Längengrad)**  
Der Längengrad „beschreibt die Ost-West-Dimension eines Ortes auf der Erdoberfläche.“ Als Ausgangspunkt dient der Nullmeridian, welcher etwa 100 Meter östlich der Londoner Sternwarte Greenwich verläuft. Positive Werte liegen östlich und negative Werte westlich des Nullmeridians. Mögliche Werte liegen zwischen -180 und +180 Grad und stellen den Winkel vom Nullmeridian zum Mess-Ort dar. [62] Dieser Wert liegt im WGS 84-Format vor.

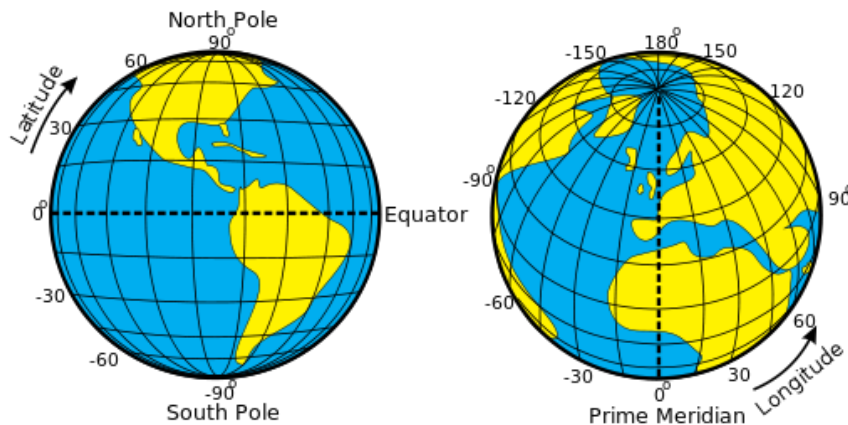


Abbildung 6.3: Längen- und Breitengrade der Erde [63]

- **HorizontalAccuracy (Horizontale Genauigkeit)**  
Wird eine Standort-Bestimmung durchgeführt, so kann die Mess-Position nicht auf den Zentimeter genau bestimmt werden. Stattdessen wird zum Längen- und Breitengrad zusätzlich ein Radius mitgegeben, in dem sich die Mess-Position ebenfalls befinden könnte. Die Horizontale Genauigkeit stellt diesen Radius in Metern dar.

Es ist zu erwähnen, dass dieser Wert nur in 68% der Fälle der Standort-Bestimmungen korrekt angegeben ist. In etwa einem Drittel liegt der wahre Punkt ausserhalb dieses Radius.[64]

- Time (Zeit)

Dieser Wert entspricht der Zeit, zu welcher die Standort-Bestimmung durchgeführt wurde. Der Wert liegt in folgendem Format vor:

JAHR-MONAT-TAG STUNDE:MINUTE:SEKUNDE.HUNDERTSTELSEKUNDE

### Wie kann ein Standort bestimmt werden?

Damit ein Smartphone den eigenen Standort bestimmen kann, gibt es die folgenden drei Möglichkeiten:

- Global Positioning System (GPS)

Das Smartphone benötigt dazu einen eingebauten GPS-Empfänger. Um die eigene Position bestimmen zu können, müssen die Bahndaten der GPS-Satelliten bekannt sein. Die Übertragung dieser Almanach-Daten geschieht entweder durch elektromagnetische Wellen, welche die Satelliten aussenden oder über eine Internetverbindung zu Referenzempfängern auf der Erde. Dabei ist die zweite Methode die schnellere.

Sobald die Almanach-Daten vorhanden sind, weiss der GPS-Empfänger, auf welche Satelliten er hören soll. Diese senden permanent ihre Zeit und Position aus. Sobald die Daten von vier Satelliten eintreffen, kann eine genaue Standort-Bestimmung durchgeführt werden. Ab diesem Zeitpunkt hat das Smartphone einen GPS-Lock. [65] Wie in den Beispieldaten der obigen Tabelle 6.1 ersichtlich ist, war dies zu jedem Zeitpunkt der Messung der Fall.

Mittels GPS kann ein Standort im besten Fall auf ca. 5 Meter genau bestimmt werden. Störend sind dabei Hindernisse wie zum Beispiel Hauswände.

- Wireless Local Area Network (WLAN)

Als Google-Mitarbeiter für das Google-Street-View-Projekt mit speziell präparierten Autos Foto-Aufnahmen der Umgebung machten, sammelten sie gleichzeitig Daten von WLAN-Routern. Sobald ein solcher erkannt wurde, merkte sich eine entsprechende Software die weltweit eindeutige Hardware-Identifikationsnummer, die Media-Access-Control (MAC)-Adresse. Diese Adresse wurde mit den dazugehörigen GPS-Koordinaten verknüpft. [66]

Führt nun ein Smartphone eine Positionsbestimmung durch, dann kann ein kürzerer Prozess durchlaufen werden, als jener bei GPS. Es genügt dabei, alle dem Gerät sichtbaren MAC-Adressen der WLAN-Router an Google zu übertragen. Es findet eine Übersetzung in GPS-Koordinaten statt, wodurch bestimmt werden kann, an welchem Ort sich das Smartphone befindet. Je mehr Router sichtbar sind, desto genauer fällt die Positionsbestimmung aus. [66][67]

Ausserhalb von Gebäuden kann ein Standort auf bis zu 10-20 Meter genau lokalisiert werden, innerhalb auf bis zu 3 Meter. [68] Damit die WLAN-Router Daten immer aktuell bleiben, werden während jeder Messung auf einem Android-Gerät die in der Umgebung gefundenen Daten an Google übertragen. [69]

- Mobilfunk-Zelle

Damit ein Mobiltelefon Telefonanrufe empfangen und tätigen kann, muss es mit dem Mobilfunknetz verbunden sein. Dieses ist in Mobilfunkzellen aufgeteilt. Da von diesen Zellen die Positionen bekannt sind und ein Mobiltelefon sich in jeweils einer Zelle befindet, kann eine Standort-Bestimmung durchgeführt werden. Wie genau diese ausfällt, hängt von der Grösse der Zelle ab. Der Wert schwankt zwischen 100 bis 500 Metern in Städten und 10 Kilometern auf dem Land. [70]

## 6.2.2 Nicht verwendete Daten

Die nachfolgenden Daten würden ebenfalls zur Verfügung stehen, werden im ausgearbeiteten Algorithmus aber nicht verwendet:

- MeasurementID (Mess-Identifikationsnummer)

Jede Messung besitzt eine eindeutige Identifikationsnummer, die MeasurementID. Diese wird nicht verwendet, da auf dem Server keine Historie über bereits angefragte Messungen angelegt wird.

- ID (Identifikationsnummer)

Die Positionen pro Messung werden durch diese Zahl von 1 bis 8 unterschieden. Der Server verwendet zur Bestimmung der korrekten Positions-Reihenfolge aber die dazugehörigen Zeiten, weshalb diese Nummer nicht benötigt wird.

- Phase (Phase)

Dieser Wert entspricht der textuellen Repräsentation der IDs, weshalb ebenfalls er nicht verwendet wird.

- Altitude (Höhenlage)

Erfolgt die Positionsbestimmung über GPS, so kann nebst dem Standort, auch die aktuelle Höhenlage bestimmt werden. Das Null-Level entspricht der Definition gemäss WGS 84, welche die Erdoberfläche durch einen Ellipsoid, also eine dreidimensionale Ellipse, vereinfacht. [71][72]

Ist bei altitude, longitude oder latitude eine 0 eingetragen, so konnte dieser Wert nicht ermittelt werden.

Im Laufe dieser Arbeit konnte kein Mehrwert durch die Verwendung dieses Parameters gefunden werden. Aus diesem Grund wird er nicht verwendet.

- Vertical Accuracy (Vertikale Genauigkeit)

Dieser Wert beschreibt die Genauigkeit der Höhenangabe als Radius in Metern. Zusammen mit der horizontalen Abweichung ergibt sich daraus eine Kugel, in welcher sich ein Messpunkt befinden kann.

Ein Wert von -1 bedeutet, dass keine Vertical Accuracy ermittelt werden konnte.

Da die Höhenlage nicht verwendet wird, gilt dies auch für die dazugehörige Genauigkeitsangabe.

- GPSLock (GPS Erhalt)

Wie im Abschnitt 6.2.1 beschrieben, bedeutet GPS-Lock, dass das Signal von mindestens vier GPS-Satelliten empfangen wird. Dies wird bei den Input-Daten mit einer 1 gekennzeichnet, im Gegensatz zu einer -1 bei nicht vorhandenem Lock.

Die zum Zeitpunkt der Arbeit bestehende Version der onlab App kann diesen Wert nicht korrekt ermitteln, weshalb er nicht verwendet wird.

### **6.2.3 Prüfung der Input-Daten**

Bevor die ankommenden Daten verarbeitet werden, wird überprüft, ob diese im erwarteten Format vorliegen. Da es sich dabei um JSON-Daten handelt, wurde für die Prüfung „JSON-Schema“ verwendet. [73] Dabei wird eine Beschreibung erstellt, wie die ankommenden Daten aufgebaut sein müssen. Diese Beschreibung wird Schema genannt. Ein Validator vergleicht anschliessend die erhaltenen Daten gegen das Schema.

Für die Daten, bei denen ein Tagging erstellt werden soll, wurde folgendes Schema festgelegt:

```

1
2 //JSON-Objekte bestehen jeweils aus Name:Value-Paaren
3
4 {
5   type: 'object',           //Es wird ein JSON-Objekt erwartet.
6   properties: {            //Dieses ist wie folgt aufgebaut:
7     positions: {           //positions ist der erste Name.
8       type: 'array',       //Dessen Value ist ein Array,
9       minItems: 8,         //welches genau acht Elemente beinhaltet.
10      maxItems: 8,
11      required: true,      //positions ist zwingend mitzugeben.
12      items: {             //Definition der einzelnen Elemente
13        type: 'object',     //Jedes Element ist ein JSON-Objekt.
14        required: true,     //Elemente mitzugeben ist zwingend.
15        properties: {
16          //Ein Element muss eine longitude vom Typ Nummer haben.
17          longitude: {
18            type: 'number',
19            required: true
20          },
21          //Ein Element muss eine latitude vom Typ Nummer haben.
22          latitude: {
23            type: 'number',
24            required: true
25          },
26          //Ein Element muss eine horizontalAccuracy vom Typ Nummer haben.
27          horizontalAccuracy: {
28            type: 'number',
29            required: true
30          },
31          //Ein Element muss eine time vom Typ String haben.
32          //(JSON kennt keinen Datentyp fuer Zeit.)
33          time: {
34            type: 'string',
35            required: true
36          }
37        }
38      }
39    }
40  }
41 }

```

Abbildung 6.4: JSON-Schema für ankommende Tagging-Daten

Nachfolgend sind die Beispieldaten aus dem Use Case (siehe 6.1) im korrekten Format aufgeführt:

```
1 {
2   "positions": [
3     {
4       "longitude": 8.7095882,
5       "latitude": 47.3589998,
6       "horizontalAccuracy": 800,
7       "time": "2017-03-28 07:31:44.0"
8     },
9     {
10      "longitude": 8.7095882,
11      "latitude": 47.3589998,
12      "horizontalAccuracy": 800,
13      "time": "2017-03-28 07:31:44.0"
14    },
15    {
16      "longitude": 8.7095882,
17      "latitude": 47.3589998,
18      "horizontalAccuracy": 800,
19      "time": "2017-03-28 07:31:44.0"
20    },
21    {
22      "longitude": 8.7135701,
23      "latitude": 47.3530638,
24      "horizontalAccuracy": 98.4000015258789,
25      "time": "2017-03-28 07:31:54.0"
26    },
27    {
28      "longitude": 8.7135701,
29      "latitude": 47.3530638,
30      "horizontalAccuracy": 98.4000015258789,
31      "time": "2017-03-28 07:31:54.0"
32    },
33    {
34      "longitude": 8.7165203,
35      "latitude": 47.3516764,
36      "horizontalAccuracy": 82.5,
37      "time": "2017-03-28 07:32:06.0"
38    },
39    {
40      "longitude": 8.7165203,
41      "latitude": 47.3516764,
42      "horizontalAccuracy": 82.5,
43      "time": "2017-03-28 07:32:06.0"
44    },
45    {
46      "longitude": 8.7165203,
47      "latitude": 47.3516764,
48      "horizontalAccuracy": 82.5,
49      "time": "2017-03-28 07:32:07.0"
50  }
  ]
}
```

Abbildung 6.5: Beispiel für korrekt formatierte Input-Daten

Als Validator wird „express-jsonschema“ [74] verwendet. Bei jedem POST-Aufruf überprüft dieser, dass das definierte Schema eingehalten wurde.

Wird nun bei der ersten Position keine horizontalAccuracy mitgeschickt, erhält der Sender folgende Antwort zurück, womit er die Fehlerursache schnell lokalisieren kann.

```

1  {
2    "statusText": "Bad Request",
3    "jsonSchemaValidation": true,
4    "validations": {
5      "body": [
6        {
7          "property": "request.body.positions[0].horizontalAccuracy",
8          "messages": [
9            "is required"
10         ]
11       }
12     ]
13   }
14 }

```

Abbildung 6.6: Server-Antwort bei nicht validem Input

### 6.2.4 Selektion für die Weiterverarbeitung

Während einer Messung der cnlab App gibt es zwar vier Phasen, dabei dauern „FCT“ und „RTT“ im Normalfall aber nur wenige Millisekunden. In den Input-Daten sind solche geringen Zeitdifferenzen nicht abgebildet, da die maximale Auflösung bei Hundertstelsekunden liegt. Somit können lediglich die folgenden drei Phasen unterschieden werden:

	MeasurementID	ID	Date	Phase
1) Vor dem Download	41043445	1	2017-03-21 19:03:53.0	FCTStart
	41043445	2	2017-03-21 19:03:53.0	FCTEnd
2) Vor dem Upload	41043445	3	2017-03-21 19:03:53.0	DownloadStart
	41043445	4	2017-03-21 19:04:03.0	DownloadEnd
	41043445	5	2017-03-21 19:04:03.0	UploadStart
3) Nach dem Upload	41043445	6	2017-03-21 19:04:14.0	UploadEnd
	41043445	7	2017-03-21 19:04:14.0	RTTStart
	41043445	8	2017-03-21 19:04:15.0	RTTEnd

Abbildung 6.7: Unterscheidbare Phasen während einer Speedtest-Messung der cnlab App

Vor der Weiterverarbeitung der Input-Punkte wird von jeder dieser unterscheidbaren Phasen der beste Punkt herausgesucht. Der Auswahlprozess wird anhand der Logik der ersten Phase „Vor dem Download“ beschrieben.

1. Es stehen folgende Punkte zur Verfügung:

- (a) Punkt 1: FCTStart
- (b) Punkt 2: FCTEnd
- (c) Punkt 3: DownloadStart

Im Fall einer langen „FCT“-Phase können zwischen den ersten beiden Punkten Sekunden liegen, was jedoch sehr selten vorkommt. Die Punkte 2 und 3 liegen zeitlich immer nahe beieinander.

2. Es werden alle Punkte daraufhin untersucht, ob latitude und longitude ungleich 0 sind und ob eine valide Zeit mitgegeben wurde. Nur Punkte, welche diese Kriterien erfüllen, werden weiterverarbeitet.

3. Sollte nur 1 gültiger Punkt vorhanden sein, so wird dieser gewählt.

4. Bei mehreren gültigen Punkten, werden diese ausgehend von Punkt 3 untersucht.

- (a) Weist Punkt 2 eine kleinere horizontalAccuracy als Punkt 3 auf und liegen die beiden Punkte maximal 100 ms auseinander, so wird Punkt 2 verwendet.
- (b) Ist der Punkt 1 nochmals genauer als Punkt 2 und liegen diese nicht weiter als 100 ms auseinander, so wird Punkt 1 verwendet.

Auf diese Weise wird der beste Punkt für die Weiterverarbeitung gefunden. Da die Umgebungsabfrage für Download- und Upload-Phasen separat gemacht wird, sollen die verwendeten Koordinaten möglichst nahe an diesen Phasen liegen. Dies wird mit der Einschränkung der Zeitdifferenz erreicht.

Anhand einer ähnlichen Logik werden die Punkte für die Phasen „Vor dem Upload“ und „Nach dem Upload“ herausgesucht.

Im Beispiel 6.5 werden die Punkte 3, 4 und 6 für die Weiterverarbeitung ausgewählt.

### **Rückweisung von Input-Daten**

Der oben beschriebene Algorithmus wählt von den ankommenden Input-Daten die korrekten und exaktesten Punkte aus. Trotzdem kann es vorkommen, dass gleich mehrere Punkte sehr ungenau oder gar ungültig sind. Nachfolgend sind diejenigen Fälle aufgeführt, welche mit einem Status-Code 400 „Bad Request“, zurückgewiesen werden:

- Pro Phase weisen alle Positionen eine latitude oder longitude von 0 auf. Dies deutet darauf hin, dass das Messgerät den Standort nicht freigegeben hat.  
Zurückgegebene Fehlermeldung: *„Cannot tag invalid positions. (Multiple occurrences of longitude or latitude 0 OR invalid time strings.)“*
- Pro Phase hat keine Position einen gültigen Datums-String.  
Zurückgegebene Fehlermeldung: *„Cannot tag invalid positions. (Multiple occurrences of longitude or latitude 0 OR invalid time strings.)“*

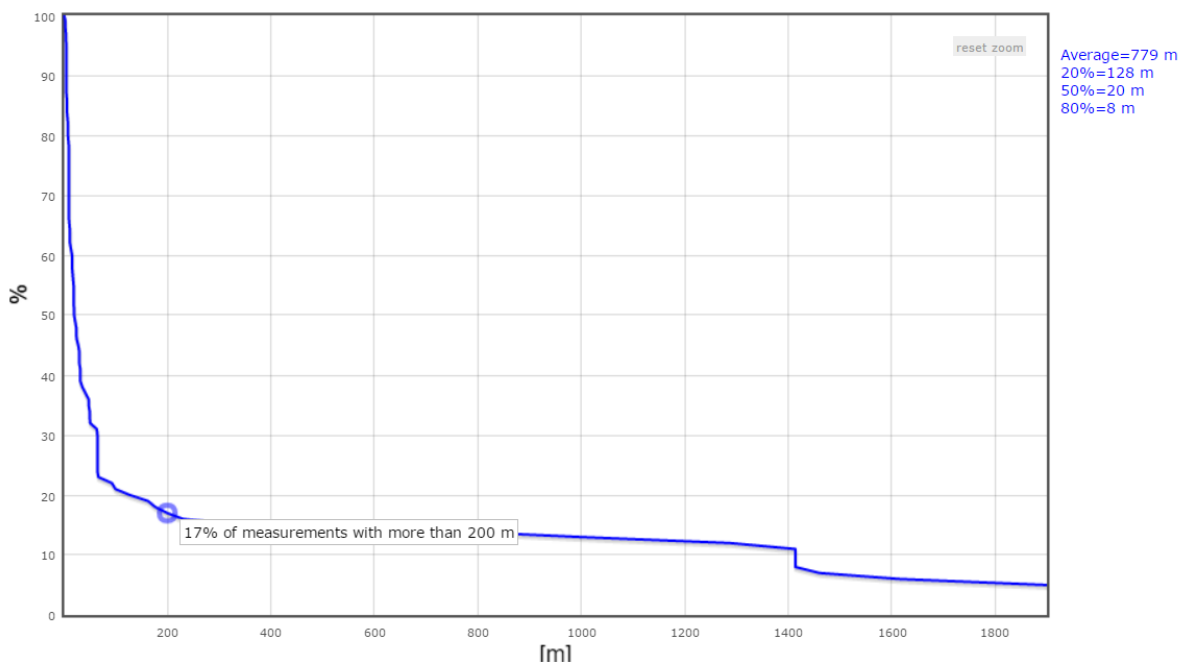
- Mindestens einer der drei ausgewählten Punkte liegt ausserhalb der Schweiz. Der implementierte Tagging-Algorithmus funktioniert jedoch nur für Daten innerhalb der Schweiz. Soll dieser auch mit Daten aus anderen Ländern eine Klassierung durchführen können, so sind unter anderem die Maximalgeschwindigkeiten bei Strassen oder die OSM-Tags für Schienen, Strassen und Gebäude zu überprüfen. Zudem muss ein anderes API zur Abfrage von Gemeindetypen und Bevölkerungsdichte gefunden werden.

Zurückgegebene Fehlermeldung: „*Not all positions are located within Switzerland.*“

- Alle der drei ausgewählten Punkte weisen eine horizontalAccuracy von mehr als 200 Metern auf.

Die Schwelle von 200 Metern wurde aufgrund der nachfolgenden Auswertung aller Einzelmessungen des letzten Jahres gewählt. Es ist ersichtlich, dass lediglich 17% der Messungen eine horizontalAccuracy von mehr als 200 Metern aufweisen. Jenseits dieser Grenze flacht die Kurve aus. Dies bedeutet, dass wenn nur 1 bis 2 % mehr Messungen angenommen werden sollten, die Akzeptanzschwelle stark erhöht werden müsste.

Zurückgegebene Fehlermeldung: „*Cannot tag positions less accurate than 200 meters.*“



Accuracy [m] MCC=228 , All Mobile, Measurement Filter: Not scheduled (Single), (24-05-2016 - 24-05-2017, Measurements: 78067, Devices: 10524)

Abbildung 6.8: Kumulative Verteilung der horizontalAccuracy aller Einzelmessungen des letzten Jahres, erstellt mit cnlab Mobile Cockpit [51]

- Alle drei ausgewählten Positionen weisen die gleiche Zeit auf.  
Zurückgegebene Fehlermeldung: „*All positions have the same time.*“

- Die drei ausgewählten Positionen liegen so weit auseinander, dass eine Geschwindigkeit von mehr als 450 km/h resultiert.  
Zurückgegebene Fehlermeldung: „*The input-positions are too far away from each other.*“

### **6.3 Output-Daten**

Werden die Input-Punkte aus dem Beispiel 6.5 an den Tagging-Server gesendet, so erhält man folgende Antwort zurück:

```
1 {
2   "title": "Calculated Tagging",
3   "location": {
4     "id": 1,
5     "name": "railway",
6     "description": "Includes OpenStreetMap-Key: railway, Values: rail,
7       light_rail, narrow_gauge, tram and subway.",
8     "weight": 0.52,
9     "allWeights": {
10      "railway": {
11        "weight": 0.52
12      },
13      "street": {
14        "weight": 0.00
15      },
16      "building": {
17        "weight": 0.00
18      }
19    },
20    "typeOfMotion": {
21      "id": 4,
22      "name": "high-speed-vehicular",
23      "description": "140 km/h to 450km/h"
24    },
25    "velocity": {
26      "distanceMeters": 996,
27      "timeSeconds": 22,
28      "velocityMeterPerSecond": 45,
29      "velocityKilometersPerHour": 163
30    },
31    "surroundings": {
32      "download": {
33        "populationDensity": {
34          "number": 73,
35          "description": "Average of persons living in 1ha based on a
36            radius-search of 300 meters."
37        },
38        "communityType": {
39          "id": 2,
40          "type": "Nebenzentrum eines Grosszentrums",
41          "description": "Tag comes from: Gemeindetypologie ARE (Bundesamt
42            fuer Raumentwicklung)",
43          "communityId": "198",
44          "communityName": "Uster",
45          "cantonId": "1",
46          "cantonName": "ZH"
47        }
48      }
49    }
50  },
51  "typeOfMotion": {
52    "id": 4,
53    "name": "high-speed-vehicular",
54    "description": "140 km/h to 450km/h"
55  },
56  "velocity": {
57    "distanceMeters": 996,
58    "timeSeconds": 22,
59    "velocityMeterPerSecond": 45,
60    "velocityKilometersPerHour": 163
61  },
62  "surroundings": {
63    "download": {
64      "populationDensity": {
65        "number": 73,
66        "description": "Average of persons living in 1ha based on a
67          radius-search of 300 meters."
68      },
69      "communityType": {
70        "id": 2,
71        "type": "Nebenzentrum eines Grosszentrums",
72        "description": "Tag comes from: Gemeindetypologie ARE (Bundesamt
73          fuer Raumentwicklung)",
74        "communityId": "198",
75        "communityName": "Uster",
76        "cantonId": "1",
77        "cantonName": "ZH"
78      }
79    }
80  }
81 }
```

```

46     "geographicalSurroundings": {
47         "osmKey": "landuse",
48         "osmValue": "residential",
49         "description": "Tag comes from: OpenStreetMap"
50     }
51 },
52 "upload": {
53     "populationDensity": {
54         "number": 64,
55         "description": "Average of persons living in 1ha based on a
                    radius-search of 300 meters."
56     },
57     "communityType": {
58         "id": 2,
59         "type": "Nebenzentrum eines Grosszentrums",
60         "description": "Tag comes from: Gemeindetypologie ARE (Bundesamt
                    fuer Raumentwicklung)",
61         "communityId": "198",
62         "communityName": "Uster",
63         "cantonId": "1",
64         "cantonName": "ZH"
65     },
66     "geographicalSurroundings": {
67         "osmKey": "landuse",
68         "osmValue": "residential",
69         "description": "Tag comes from: OpenStreetMap"
70     }
71 }
72 }
73 }

```

Abbildung 6.8: Beispiel-Antwort des Tagging-Servers

### Beschreibung

- **Location (Messort):**  
Gibt die location, deren Gewichtung sowie die Gewichtung aller möglichen locations zurück. Wie diese Werte zustande kommen ist im Abschnitt 6.4.2 beschrieben.
- **Type of motion (Bewegungsart):**  
Gibt den type of motion zurück. Die Berechnung dieses Wertes ist in Abschnitt 6.4.1 beschrieben.
- **Velocity (Geschwindigkeit):**  
Gibt die Strecke, Zeit und Fortbewegungs-Geschwindigkeit während der Messung zurück. Diese Rückgaben werden, zusammen mit dem type of motion, im Abschnitt 6.4.1 genauer beschrieben.

- Surroundings (Umgebung):

Dieser Teil der Antwort beinhaltet einerseits den im Use Case 6.1.2 beschriebenen Punkt „Population density“ sowie einen Teil des Punktes „Location“. Der Punkt „Location“ ist somit in zwei Antwort-Teilen enthalten. Diese Aufteilung wurde vorgenommen, da sich eine Person gemäss der location-Definition im Use Case an zwei Orten gleichzeitig, z.B. auf einer Strasse durch den Wald, aufhalten könnte. Somit stellt die Strasse den eigentlichen Messort, also die location, dar und der Wald die geografische Umgebung. Da die „Population density“ ebenfalls die Umgebung beschreibt, entstand daraus der Antwort-Teil „Surroundings“. Die darin enthaltenen Werte werden für Download und Upload separat berechnet und sind in den Abschnitten 6.4.3 bis 6.4.5 genauer beschrieben.

## 6.4 Logik

### 6.4.1 Velocity and type of motion

Die velocity errechnet sich aus der Durchschnittsgeschwindigkeit zwischen dem ersten und dem zweiten sowie dem zweiten und dem dritten ausgewählten Punkt. Anhand dieses Wertes wird der type of motion gemäss der Definition im Use-Case (siehe 6.1.2) abgeleitet. Die Umsetzung im Code sieht wie folgt aus:

```

1  function getTypeOfMotion(speed) {
2
3      switch(true) {
4          case (speed >= 0 && speed < 3):
5              return STATIONARY;
6          case (speed >= 3 && speed < 11):
7              return PEDESTRIAN;
8          case (speed >= 11 && speed < 140):
9              return VEHICULAR;
10         case (speed >= 140 && speed < 450):
11             return HIGH_SPEED_VEHICULAR;
12         default:
13             return UNKNOWN;
14     }
15 }

```

Abbildung 6.9: Bestimmung des type of motion

Die Variable speed kann dabei keine negativen Werte annehmen, da diese aus der Division der folgenden Werten hervorgeht:

- Distanz aus der PostGIS-Funktion „ST\_Distance [75]. Eine Distanz ist dabei immer grösser oder gleich 0.
- Absoluten Zeit zwischen zwei Punkten

## 6.4.2 Bestimmung der location

Der implementierte Algorithmus kann folgende drei locations unterscheiden:

- Railway (id: 1)  
Dieser location beinhaltet den OSM-Key „railway“, also „Eisenbahnen oder anderer Transportsysteme welche Schienen nutzen“[76]. Davon werden die folgenden Values verwendet:
  - light\_rail (Stadtbahn)
  - narrow\_gauge (Schmalspurbahn)
  - rail (Eisenbahn, S-Bahn)
  - subway (U-Bahn)
  - tram (Strassenbahn)

Diese sind unter der folgenden URL genauer beschrieben:  
<http://wiki.openstreetmap.org/wiki/DE:Key:railway>

- Street (id: 2)  
„street,“ bedeutet nicht nur Autos und Busse, sondern auch Fussgänger. Somit wird dafür der OSM-Key „highway“[77] verwendet, unter welchen Strassen, Wege und Pfade fallen. Nachfolgend sind die benutzten Values aufgelistet:
  - motorway (Autobahn)
  - trunk (Autostrasse)
  - primary (Hauptstrasse)
  - secondary (Hauptstrasse)
  - tertiary (Nebenstrasse)
  - unclassified (Nebenstrasse)
  - residential (Strasse an und in Wohngebieten)
  - service (Erschliessungsweg)
  - living\_street (Wohnstrasse)
  - track (Wirtschafts-, Feld- oder Waldweg)
  - road (Strasse ohne Zuordnung)

Hinzu kommen Auffahrten, z.B. „primary\_link“, welche ebenfalls bei OSM abgefragt werden. Weiter werden bei Geschwindigkeiten bis zu 10 km/h die folgenden Values miteinbezogen:

- pedestrian (nur für Fussgänger erlaubter Weg, Platz oder Strasse)

- footway (Fussweg)
- path (Wanderwege oder Trampelpfade)

Eine genaue Beschreibung ist unter folgender URL zu finden:  
<http://wiki.openstreetmap.org/wiki/DE:Key:highway>.

Der OSM-Key „footway“ [78] wurde nicht verwendet, da diese Einträge entweder direkt neben oder auf Strassen liegen. Dies würde dazu führen, dass eine einzelne Strasse doppelt zurückgegeben wird und „railway“ damit benachteiligt wird.

- Building (id: 3)

Wird dieser Wert zurückgegeben, so fand die Messung in oder auf einem Gebäude statt, welches gemäss Wikipedia wie folgt definiert ist: „Ein Gebäude ist ein Bauwerk, das Räume einschliesst, betreten werden kann und dem Aufenthalt von Menschen, Tieren oder der Lagerung von Sachen dient.“ [79]

Der Algorithmus sucht bei OSM nach den Einträgen bei denen „building“ eingetragen ist. Zusätzlich wird in mit dem Key „tourism“ [80] nach folgenden Values gesucht:

- hut (Hütte)
- alpine\_hut (Berghütte)
- mountain\_hut (Berghütte)
- wilderness\_hut (abgelegene Schutzhütte)
- apartment (Ferienwohnung)
- chalet (Ferienhaus)
- resort (Ferienort)
- hotel (Hotel)
- motel (Motel)
- bed\_and\_breakfast (Pension)
- guest\_house (Pension)
- restaurant (Restaurant)
- castle (Burg)
- casino (Casino)

Diese Liste basiert auf der Suche nach eingetragenen Werten der OSM-Karte der Schweiz. Doppelzählungen sind ausgeschlossen, da nur geprüft wird ob, und nicht wie viele, Gebäude in der Nähe sind.

- Unknown (id: -1)

Kann auf keine der drei obigen locations zurückgeschlossen werden, so wird unknown zurückgegeben.

Jede der locations erhält eine Gewichtung von maximal 1. Je grösser diese nach dem Durchlaufen des Algorithmus ist, desto mehr Indikatoren wurden für diese location gefunden.

Im nachfolgenden Abschnitt wird das Konzept der Punkt-Gewichtung beschrieben. Anschliessend werden die durchlaufenen Schritte zur Location-Bestimmung erläutert. Schlussendlich wird auf die Abfragen an die OSM-Datenbank genauer eingegangen.

### Punkt-Gewichtung

Dies ist ein zentrales Konzept bei der Bestimmung der Location. Damit wird erreicht, dass ungenau aufgelöste Koordinaten einen geringeren Einfluss auf das Resultat haben als genaue Koordinaten. Hierzu wird die Berechnung der Gewichtung anhand von Horizontal Accuracies dreier Beispielpunkte hergeleitet:

Punkt	HorizontalAccuracy	Gewichtung	Definitive Gewichtung
1	150		
2	100		
3	50		

Abbildung 6.10: Beispiel-Punkte

Der dritte Punkt ist doppelt so genau wie der zweite und dreimal so genau wie der erste Punkt. Ziel ist eine Gewichtung, bei dem der dritte Punkt zweimal so viel wie der zweite und dreimal so viel wie der erste Punkt zählt.

Damit für die Berechnung einfache Zahlen resultieren, wird festgelegt, dass der erste Punkt eine Gewichtung von 2 erhält. Die Gewichtungen der restlichen Punkte leiten sich aus der bisherigen Beschreibung ab. Dies ist in der nachfolgenden Tabelle dargestellt:

Punkt	HorizontalAccuracy	Gewichtung	Definitive Gewichtung
1	150	2	
2	100	3	
3	50	6	

Abbildung 6.11: Gewichtete Beispiel-Punkte

Die Gewichtung der Punkte kann mit der folgenden Formel berechnet werden.  $Punkt_i$  entspricht dabei der *HorizontalAccuracy* des Punktes  $i$ .

$$TotalHorizontalAccuracy = Punkt_1 + Punkt_2 + Punkt_3$$

$$Gewichtung_i = \frac{TotalHorizontalAccuracy}{Punkt_i}$$

Da die schlussendliche Gewichtung aller Punkte 1 betragen soll, wird folgende Gleichung aufgestellt:

$$\begin{aligned} 2x + 3x + 6x &= 1 \\ 11x &= 1 \\ x &= 0,0909090909090909 \end{aligned}$$

Daraus folgt :

$$\begin{aligned} x &= 1 / 11 \\ x &= 1 / (\text{Gewichtung}_1 + \text{Gewichtung}_2 + \text{Gewichtung}_3) \end{aligned}$$

Das gefundene x kann nun mit der Gewichtung pro Punkt multipliziert werden, um die definitive Gewichtung zu erhalten.

$$\begin{aligned} \text{DefGewicht}_3 &= 6 * x \\ \text{DefGewicht}_3 &= 6 * 0,0909090909090909 \\ \text{DefGewicht}_3 &= 0,5454545454545455 \end{aligned}$$

Somit erfolgt durch Einsetzen die schlussendliche Formel:

$$\text{DefGewicht}_i = \text{Gewichtung}_i * x$$

$$\text{DefGewicht}_i = \text{Gewichtung}_i * 1 / (\text{Gewichtung}_1 + \text{Gewichtung}_2 + \text{Gewichtung}_3)$$

$$\text{DefGewicht}_i = \text{Gewichtung}_i * 1 / \left( \frac{\text{TotalHorizontalAccuracy}}{\text{Punkt}_1} + \frac{\text{TotalHorizontalAccuracy}}{\text{Punkt}_2} + \frac{\text{TotalHorizontalAccuracy}}{\text{Punkt}_3} \right)$$

$$\text{DefGewicht}_i = \frac{\text{TotalHorizontalAccuracy}}{\text{Punkt}_i} * 1 / \left( \frac{\text{TotalHorizontalAccuracy}}{\text{Punkt}_1} + \frac{\text{TotalHorizontalAccuracy}}{\text{Punkt}_2} + \frac{\text{TotalHorizontalAccuracy}}{\text{Punkt}_3} \right)$$

Vereinfacht:

$$\text{DefGewicht}_i = \frac{\text{TotalHorizontalAccuracy}}{\text{Punkt}_i} / \left( \frac{\text{TotalHorizontalAccuracy}}{\text{Punkt}_1} + \frac{\text{TotalHorizontalAccuracy}}{\text{Punkt}_2} + \frac{\text{TotalHorizontalAccuracy}}{\text{Punkt}_3} \right)$$

HorizontalAccuracy	Gewichtung	Definitive Gewichtung
150	2	0,181818
100	3	0,272727
50	6	0,545454

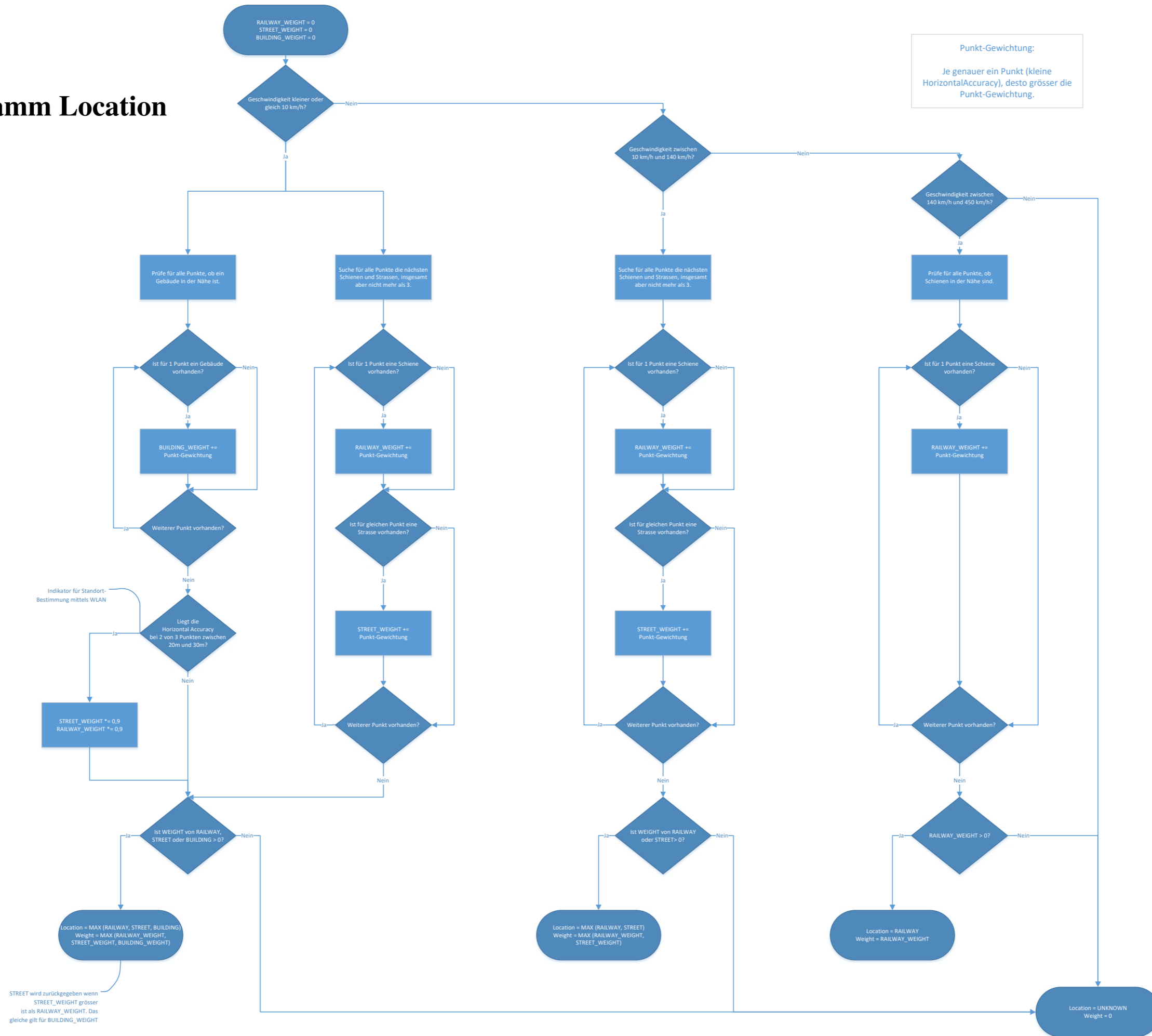
Abbildung 6.12: Definitive Gewichtung der Beispiel-Punkte

Die Funktion „getPositionWeight“ aus der Datei „location.js“ entspricht der Implementation dieser Formel.

**Ablauf**

Das nachfolgende Flussdiagramm zeigt die Schritte zur Bestimmung einer location auf:

# Flussdiagramm Location



Punkt-Gewichtung:  
Je genauer ein Punkt (kleine HorizontalAccuracy), desto grösser die Punkt-Gewichtung.

Zu Beginn werden WEIGHT von railway, street und building auf null gesetzt. Diese werden während des Durchlaufens des Algorithmus erhöht, sobald ein Indiz für die entsprechende location vorliegt.

Der erste Indikator ist die Geschwindigkeit der Messperson. Bis zu 10 km/h kann noch keine der locations ausgeschlossen werden, weshalb in diesem Fall alle überprüft werden müssen. Bei Messungen über dieser Geschwindigkeit kann building ausgeschlossen werden. Schliesslich kann ab 140 km/h auch street ausgeschlossen werden, da die erlaubte Maximalgeschwindigkeit auf Strassen in der Schweiz bei 120 km/h liegt.

Der Fall ab 140 km/h, bei dem nur noch überprüft werden muss, ob die Messung im Zug stattfand, stellt den einfachsten Fall dar. Auf diesen wird deshalb zuerst eingegangen.

Als erster Schritt wird die unten aufgeführte Datenbankabfrage 6.13 für alle drei ausgewählten Punkte ausgeführt. Diese liefert für jeden Punkt zurück, ob sich innerhalb von 20 Metern eine Zugstrecke befindet. Ist dies der Fall, so wird die Gewichtung des entsprechenden Punktes zur RAILWAY\_WEIGHT dazugezählt. Nachdem alle Punkte durchlaufend sind, wird überprüft, ob irgendeine Zugstrecke gefunden wurde. Trifft dies zu, wird die location railway mit der berechneten RAILWAY\_WEIGHT zurückgegeben, ansonsten unknown mit einer Gewichtung von null.

Bei der Geschwindigkeit zwischen 10 und 140 km/h werden mit einer Datenbankabfrage nach den nächsten Strassen und Schienen, im Umkreis von 10 Metern, gesucht. Maximal soll das Resultat pro Punkt drei Rückgaben enthalten. Anschliessend wird bei jedem Punkt untersucht, ob Strassen oder Schienen gefunden wurden, was beim Vorhandensein dazu führt, dass das entsprechende WEIGHT erhöht wird. Am Ende wird diejenige location mit dem grösseren WEIGHT zurückgegeben oder, falls beide null sind, unknown.

Übrig bleibt der Pfad, bei dem durch die Geschwindigkeit keine location ausgeschlossen werden konnte. Dabei ist die Prüfung, ob Strassen oder Schienen in der Nähe sind, genau die gleiche wie bei der Geschwindigkeit zwischen 10 und 140 km/h. Der Ablauf für building entspricht jenem für railways, jedoch mit einer Ausnahme. Beim manuellen Tagging wurde festgestellt, dass die Messungen in Gebäuden grösstenteils eine horizontalAccuracy von 20 bis 30 Metern aufwiesen. Diese Beobachtung wird im location-Algorithmus widerspiegelt, indem bei dieser Auflösungsgenauigkeit die WEIGHTS von Strasse und Schiene herabgesetzt werden. Auch hier wird am Schluss geprüft, welche location den grössten WEIGHT hat und diese wird dann zurückgegeben.

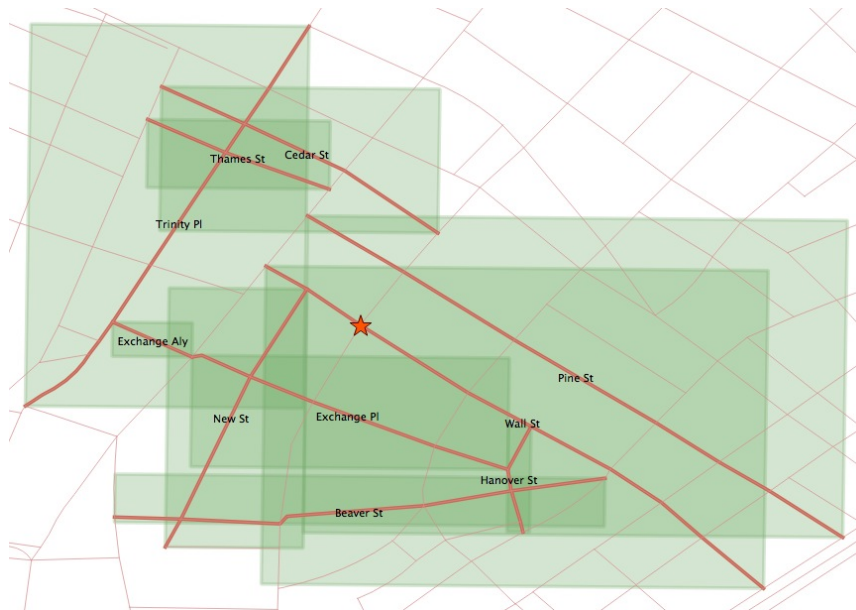
### **Datenbankabfragen am Beispiel von railway**

Die Abfrage hat den Zweck, die OSM-ID einer Zugstrecke zurückzugeben, welche innerhalb von 10 Metern zur mitgegebenen Koordinate liegt. Da sowohl die OSM-Daten wie auch die Messungen der cnlab App als WGS 84-Koordinaten vorliegen, ist keine Konvertierung nötig. [81]

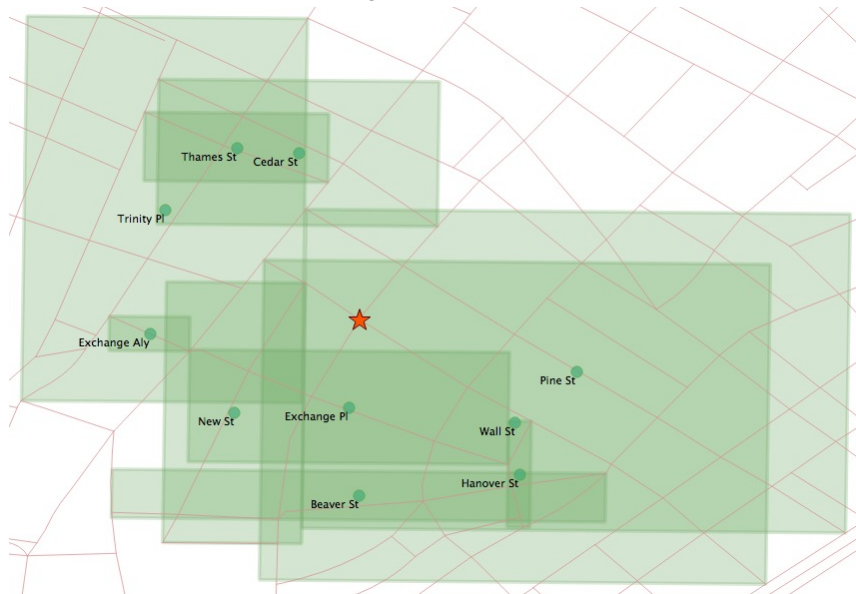
```
1 WITH closest_candidates AS (  
2     SELECT osm_id, way  
3     FROM planet_osm_line  
4     WHERE railway  
5         IN ('rail', 'light_rail', 'narrow_gauge', 'tram', 'subway')  
6     ORDER BY way <-> ST_GeomFromText('POINT(8.71157915 47.3560318)', 4326)  
7     LIMIT 100  
8 )  
9 SELECT osm_id  
10 FROM closest_candidates  
11 WHERE ST_Distance(way::geography,  
12     ST_GeomFromText('POINT(8.7165203 47.3516764)', 4326)::geography) < 20  
13 LIMIT 1;
```

Abbildung 6.13: Nearest-Neighbour Searching für Railways

Damit eine performante Abfrage möglich ist, wird diese in zwei Schritten durchgeführt. Der erste Schritt ist die „Index-based KNN (K nearest neighbours) Suche“, also die Suche nach den K nächsten Objekten, welche sich den Spatial-Index (siehe Abschnitt 6.5.2) zu Hilfe nimmt. Dazu wird der „<->“-Operator verwendet. Da aber dieser Index aufgrund von Bounding Boxen aufgebaut ist, entsprechen die Suchresultate nicht den wahren, nächsten Objekten, wie auf dem folgenden Bildern beispielhaft dargestellt ist:



(a) Die Such-Koordinate ist als Stern, die nächsten Strassen sind orange eingezeichnet.[82]



(b) Nächste Bouding Boxen mit ihrem Mittelpunkt in grün [82]

gid	name	distance
17360	Exchange Pl	101.6241843136
17350	New St	63.9499165490674
17385	Wall St	0.714202224374917

(c) Resultat der Index-based KNN-Suche [82]

Es ist ersichtlich, dass, obwohl der Messpunkt direkt auf der Wall-Street liegt, die Index-

based KNN-Suche etwas anderes zurückgibt. Der Grund ist in Abbildung (b) zu erkennen. Diese stellt die Mittelpunkte der Bouding Boxen dar, nach welchen das Resultat sortiert wurde.

Mittels Index-based KNN können also auf eine schnelle Art die umliegenden Objekte bestimmt werden, diese können in Wahrheit aber eine grosse Distanz zur Such-Koordinate aufweisen. Aus diesem Grund wird im zweiten Teil der Abfrage der Abstand mittels der PostGIS-Funktion „ST\_Distance“ [75] berechnet, welche die minimale Distanz zwischen zwei Geometrien zurückgibt.

### **Korrektheit der berechneten location**

Mit dem final umgesetzten Algorithmus konnte folgende Korrektheit erreicht werden:

- location railway: 72%
- location street: 92%
- location building: 82%
- über alle location: 82%

Die detaillierten Auswertungen können im Kapitel 14.3.2 „Auswertung aufgrund des manuellen Taggings“ eingesehen werden.

### **6.4.3 Bestimmung des community type**

Der Gemeindetyp wird beim GeoAdmin API [1] mittels des Services „ch.are.gemeindetypen“ abgefragt. Dieser liefert jeweils den „Gemeindetyp ARE“ zurück, welcher wie folgt definiert ist:

„Die Gemeindetypen ARE sind das Ergebnis einer Kombination zwischen den Grossregionen, der Agglomerationsdefinition 2000 sowie der Gemeindetypologie des Bundesamts für Statistik BFS. Aus den ursprünglich 13 Typen wurde eine Typologie bestehend aus 9 Typen abgeleitet.“ [83] [84]

Dabei sind die folgenden Gemeindetypen definiert:

- Grosszentrum
- Nebenzentrum eines Grosszentrums
- Gürtel eines Grosszentrums
- Mittelzentrum
- Gürtel eines Mittelzentrums
- Kleinzentrum
- Periurbane ländliche Gemeinde

- Agrargemeinde
- Touristische Gemeinde

Da einer Messung jeweils nur einer Gemeinde zugewiesen werden soll, wird bei der Abfrage ein Suchradius von 0 Metern spezifiziert. Der entsprechende Parameter heisst „tolerance“. Gibt die API-Abfrage als Resultat ein leeres Array zurück, so lag die mitgelieferte Position wahrscheinlich innerhalb eines grösseren Sees. Bei diesen sind keine Gemeinden und damit auch keine Gemeindetypen eingetragen, wie auf dem folgenden Bild beim Greifensee ersichtlich ist:

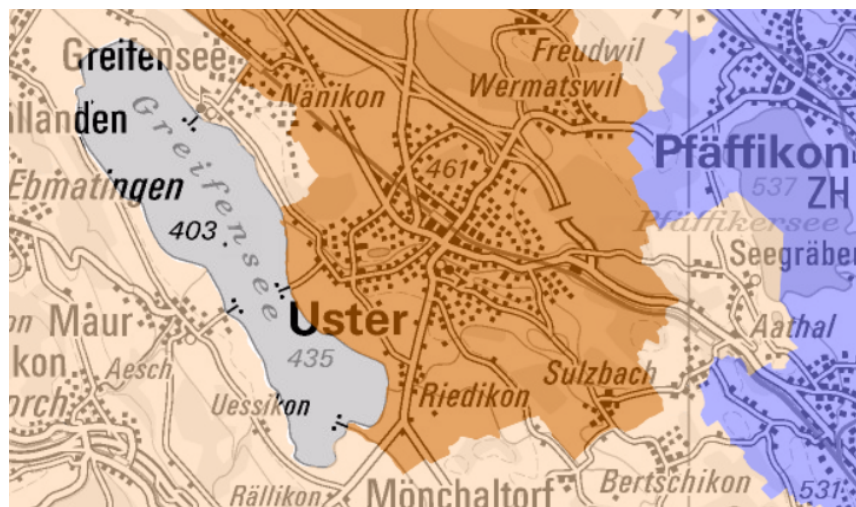
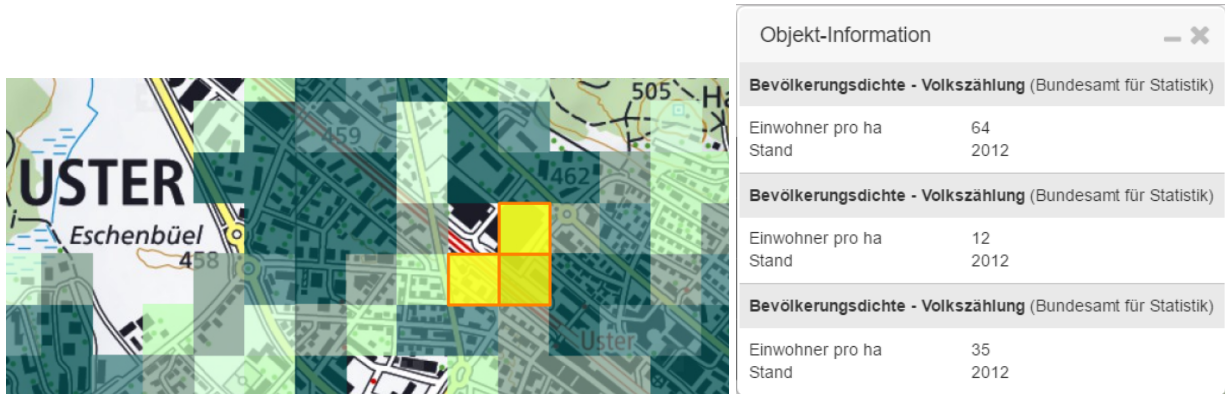


Abbildung 6.15: Gemeindetypen in den Farben hellbraun (Gürtel eines Grosszentrums), braun (Nebenzentrum eines Grosszentrums) und dunkelblau (Mittelzentrum).[83]

#### 6.4.4 Bestimmung der population density

Zur Bestimmung der Besiedlungsdichte wird der Service „ch.are.bevoelkerungsdichte“ des GeoAdmin API [1] verwendet. Dessen Daten stammen von den Personenregistern des Bundes sowie den Einwohnerregistern der Gemeinden und Kantone[85].

Die verwendete Einheit ist dabei Anzahl Personen pro Hektare. Dabei kommt es vor, dass zwei nebeneinanderliegende Felder von 100 à 100 Meter stark abweichende Bevölkerungsdichten aufweisen. Um dies zu umgehen, wird in der Anfrage, neben dem Längen und Breitengrad, ein Radius von 300 Metern definiert. Zudem soll damit die Auslastung einer Mobilfunkzelle abgebildet werden, deren wahre Grösse allerdings nicht bekannt ist. Falls zu einem späteren Zeitpunkt die Zellengrösse herausgefunden werden kann, sollten die 300 Meter durch die effektive Zellengrösse ersetzt werden. Der entsprechende Parameter in der URL heisst „tolerance“. Die damit erhaltene Antwort enthält keine bis mehrere Treffer. Ersteres ist der Fall, wenn am angefragten Ort keine Personen wohnen. Bei mehreren Treffern wird dann der Durchschnitt der Bevölkerungsdichten berechnet und zurückgegeben.



(a) Abfrage der Bevölkerungsdichte am Bahnhof Uster [86]

(b) Antwort des GeoAdmin API zu den drei im Bild (a) gelb gekennzeichneten Kästchen [86]

### 6.4.5 Bestimmung der geographicalSurroundings

Die geografische Umgebung wird aus den Daten von OpenStreetMap hergeleitet. In der Datenbankabfrage wird nach den folgenden OpenStreetMap-Keys gesucht:

- **Boundary (Grenzen)**  
Mit boundary werden Gebiete gekennzeichnet, welche eine politische oder administrative Bedeutung haben [87]. Hier wird bewusst der Value administrative ausgeschlossen, denn in diesem befinden sich sämtliche Gemeinde-, Kantons- und Landesgrenzen. Diese Grenzen werden für die Anzeige der geografischen Umgebung nicht benötigt.
- **Natural (natürlich)**  
Der Key natural „beschreibt physische und geologische Landschaftselemente sowie natürliche Vegetationsformen.“ [88]
- **Leisure (Freizeit)**  
Dieser Key „beschreibt Orte, an denen Menschen ihre Freizeit verbringen.“ [89]
- **Landuse (Bodennutzung)**  
Mit dem Key landuse wird die überwiegende Nutzung eines Gebietes beschrieben. [90]

Es kann sein, dass bei einem Punkt für mehrere der oben genannten Keys ein Datenbankeintrag besteht. Da nur ein Eintrag zurückgegeben wird, wurde folgende Priorisierung festgelegt:

- **Priorität 1:** landuse
- **Priorität 2:** leisure
- **Priorität 3:** natural
- **Priorität 4:** boundary

In der oben stehenden Aufzählung wurde die Priorität nach Wichtigkeit angegeben, das heisst die Priorität 1 wird zurückgegeben, falls sie vorhanden ist. Nur falls kein Priorität 1 Eintrag vorhanden ist, kommt ein Eintrag aus Priorität 2 oder tiefer zurück.

## 6.5 Performance-Optimierungen

### 6.5.1 Parallel ablaufender Code

Da das Berechnen der verschiedenen Rückgabewerte zu grossen Teilen unabhängig voneinander funktioniert, können Code-Abschnitte parallel zueinander ablaufen. Als Beispiel ist unten ein Auszug aus der Datei „taggingCommunication“ aufgeführt. Es ist ersichtlich, dass durch die parallele Ausführung die Web-Anfragen an das GeoAdmin-API nicht auf die Datenbankresultate der ersten beiden Funktionen warten muss.

```

1  parallel([
2    function(callback) {
3      location.getLocation(...); //Datenbankaufruf
4    },
5    function(callback) {
6      geographicalSurroundings.getGeographicalSurroundings(...);
7      //Datenbankaufruf
8    },
9    function(callback) {
10     populationSurroundings.getGeoAdminData(...) //externer Web-API-
11       Aufruf
12     },
13     function(err, results) {...});

```

Abbildung 6.17: Parallele Aufrufe in der Funktion renderTagJSON

Für die parallelen Aufrufe wird die Funktion „parallel“ aus dem npm-Modul „async“ [91][92] verwendet. Diese nimmt als Argumente eine Liste von asynchronen Funktionen sowie eine Callback-Funktion entgegen. Die Callback-Funktion ist in der Abbildung 6.17 auf Zeile 13 ersichtlich. Diese Funktion wird aufgerufen, wenn alle asynchronen Funktionen durchgelaufen sind. Darin werden die erhaltenen Resultate verarbeitet und zurückgegeben.

### 6.5.2 Optimierung der Datenbank

Trotz des parallel ablaufenden Codes konnte lediglich eine Antwortzeit von ca. 4 Sekunden erreicht werden. Vor allem die Datenbankabfragen dauerten lange, weshalb folgende Techniken angewandt wurden [93]:

- **Spatial Indexing**

Erstellt pro Tabelle einen räumlichen Index aufgrund der Bounding Boxen der Einträge.

Dieser hilft dabei, die Nearest Neighbour Search (siehe. 6.13) zu beschleunigen. Der Index wird beim Datenbank-Import durch osm2pgsql [94] automatisch erstellt.

- **Vacuuming**

Aufräumen von nicht mehr benötigten Datenbankeinträgen. Dies ist nach vielen UPDATE- oder DELETE-Operationen der Fall. [95]

- **Analyzing**

Erneuert die internen Statistiken für die Ausführungspläne. Diese beinhalten Strategien, wie PostgreSQL die Einträge in bestimmten Situationen durchläuft. Diese Operation wird jeweils zusammen mit dem Vacuuming nach den Updates durchgeführt.

- **Clustering**

Sortiert die Daten aufgrund des Spatial Index. Dies führt dazu, dass bei Abfragen die Anzahl der in den Hauptspeicher geladenen Daten reduziert werden kann. Diese Operation wird direkt nach dem Datenbank-Import durchgeführt. Bei den Updates wird auf erneutes Clustering verzichtet, da während dieser Zeit keine Abfragen beantwortet werden können. Sollten die Antwortzeiten der Datenbank jedoch stark ansteigen, kann ein weiteres Clustering durchgeführt werden, was bei der aktuellen OSM-Datenbank der Schweiz ca. 30 Sekunden dauert.

Aufgrund dieser Datenbankoptimierungen liegt die Antwortzeit zum Zeitpunkt der Abgabe bei ca. 250 ms. Dabei benötigen die in Abbildung 6.17 aufgeführten Funktionen folgende Zeiten (Angaben schwanken je nach Anfrage):

- getLocation: ca. 20 ms
- getGeographicalSurroundings: ca. 15 ms
- getGeoAdminData: ca. 240ms (externer Web-API-Aufruf)

# Kapitel 7

## Weiterführende Arbeiten

### 7.1 Arbeiten am Tagging-Server

#### 7.1.1 Mögliche Verbesserungen des location-Algorithmus

- Durch die Auswertung der falsch getaggten Messungen wurden einige Verbesserungspunkte gefunden. Diese sind im Abschnitt 14.3, zusammen mit möglichen Lösungsansätzen, aufgeführt.
- Überprüfung, ob die von der Datenbank zurückgegebenen buildings alle dieselben sind und wenn nicht, die Gewichtung der buildings herabsetzen. Bringt dies alleine keine Verbesserung, so ist zudem der Suchradius von Gebäuden sowie die Anzahl zurückgegebener Gebäude zu überarbeiten.
- Überprüfung, ob die erreichte Fortbewegungsgeschwindigkeit grösser ist als die gesetzlich festgelegte Maximalgeschwindigkeit. Dies wäre ein starker Indikator für die location „railway“. Der entsprechende OSM-Key heisst „maxspeed“ [96]. Gemäss Herr Keller ist dieser allerdings erst bei ca. 30% der Schweizer Strassen eingetragen. Um den Tag zu nutzen, ist das Style-File von osm2pgsql [97] um folgenden Eintrag zu ergänzen [98][99]:

```
1 node,way maxspeed text linear
```

- Beim manuellen Tagging gab es zum Teil Messungen, bei denen sich ein Teil der Punkte in Bern und ein anderer Teil in Davos befand. Dabei handelt es sich sehr wahrscheinlich um Zug-Messungen, welche mit dem Zug-WLAN verbunden waren. Dabei erfasste Google die MAC-Adresse der WLANs zum Beispiel am Bahnhof Davos, der Zug befindet sich aber zum Zeitpunkt der aktuellen Messung in Bern. Diese Vermutung ist zu überprüfen.
- Die Schweizerische Bundesbahnen (SBB) bietet folgende Webseite an, auf denen die aktuellen Positionen von Zügen angezeigt und abgefragt werden können:  
<http://maps.trafimage.ch/#/ch.sbb.netzkarte?baselayers=ch.sbb.netzkarte.relieef, ch.sbb.netzkarte.luftbild.group&layers=ch.sbb.tracker>

Damit könnte überprüft werden, ob sich zum Zeitpunkt der Messung ein Zug am Messort befindet. Es gibt allerdings zwei grosse Probleme, welche diese Algorithmus-Optimierung verhindern. Zum einen gibt die SBB ihre Verspätungen nicht in Echtzeit bekannt. Gemäss Herr Keller tut sie dies erst am Folgetag. Zum anderen werden die Informationen nur für den privaten Gebrauch zur Verfügung gestellt.

Sollten sich diese Situationen ändern, könnte der location-Algorithmus mit Fahrplan-Informationen erweitert werden.

- Überprüfung, ob bzw. wie die gefundenen OSM-Ways aus der Datenbankabfrage zusammengehören. Bilden z.B. alle zusammen eine Strasse, so wäre dies ein Indikator für die location „street“. Um dies umzusetzen kann das Tool osm2po [100] verwendet werden, welches zu Beginn dieser Arbeit ausprobiert wurde. Die verwendete Konfigurationsdatei ist im Abgabeordner unter „5\_Realisierung\0\_Archiv\InstallationUndUpdates\Ursprüngliche Anleitung\Zusatz-Dateien\routing“. Der folgende Befehl erstellt dann im Ordner „ch“ ein SQL-File, welches in die Datenbank importiert werden kann:

```
1 java -jar osm2po-core-5.1.0-signed.jar cmd=c prefix=ch \
2 switzerland-latest.osm.pbf
```

Abbildung 7.1: OSM-Daten routingfähig machen

- Verbesserungen durch bestehende Sensor-Messungen, welche durch die cnlab App gemacht werden. Wie im Abschnitt 6.2.2 „Nicht verwendete Daten“ beschrieben, gibt es unter anderem die altitude oder den GPSPlock, welcher noch nicht ausgewertet werden. Der GPSPlock ist derzeit jedoch nicht korrekt implementiert. Des Weiteren erfasst die App weitere Werte, welche ebenfalls zur Verbesserung beitragen könnten. Bei diesen Sensor-Messungen ist zu untersuchen, welche Differenzen sie bei Zug-, Strassen- und Gebäude-Messungen aufweisen und die Erkenntnisse anschliessend in den location-Algorithmus zu übernehmen.
- Durch Messungen von zusätzlichen Sensoren können weitere Unterschiede zwischen den Messorten aufgedeckt werden. Möglich sind zum Beispiel die folgenden:
  - Hotspots: Die Art der sichtbaren SSID könnte Hinweis auf Zug-, Strassen- und Bahnhofsumgebung liefern.
    - \* **Zugs Umgebung:** Smartphone Manufacturer Codes mit hohem Signalpegel
    - \* **Strassen Umgebung:** typische Access Point wie z.B. UPC\_xy oder Fritz!Box\_xy mit unterschiedlichen Signalpegeln
    - \* **Bahnhofsumgebung:** typische Access Point, wie z.B. SBB-Free oder MCDonalds's-Free mit unterschiedlichen Signalpegeln

Die Vielzahl der SSIDs gibt zudem einen Hinweis auf eine ländliche oder städtische Umgebung.

- Bewegungssensor: In der Schweiz weisen die Eisenbahnschienen häufig mehr Unebenheiten auf als die Strassen.
- Beschleunigungssensor: Die Geschwindigkeit im Zug nimmt gleichmässiger zu als im Auto.
- Der community type sowie die population density werden derzeit beim GeoAdmin API abgefragt. Die Bestimmung dieser Werte dauert ca. zehn Mal so lang wie jene der Werte, welche anhand von Datenbankabfragen berechnet werden. Diese Lösung wurde gewählt, da die Daten der Gemeindetypen [101] nach dem Datenbank-Import nicht verwendet werden konnten. Der Grund war, dass die Koordinaten nicht korrekt erkannt wurden. Bei der Bevölkerungsdichte stehen zudem nur Excel-Dateien [102] zur Verfügung. Diese müssten zuerst aufbereitet werden, liefern dann aber immer noch weniger Daten als die Online-Abfrage zurück.  
Sollten seitens des Bundes bessere Daten zur Verfügung stehen, so können diese in die lokale Datenbank übernommen und von dort aus abgefragt werden.
- Soll der Tagging-Server auch mit Daten aus anderen Ländern umgehen können, so sind folgende Teile des Algorithmus zu überarbeiten bzw. zu ändern:
  - Geschwindigkeitsgrenzen zur Bestimmung des type of motion
  - OSM-Tags zur Bestimmung von Zugstrecken und Strassen
  - Datenquelle für den Bezug von community type und population density
  - Datenbank mit den OSM-Daten des entsprechenden Landes

### 7.1.2 Weitere Verbesserungen

- Statt der Gewichtung pro zurückgegebener location soll eine probability über alle locations berechnet werden, welche zusammengerechnet immer 1 ergeben. Diese zeigt an, wie sicher das Resultat bestimmt werden konnte.
- Bei der Erstellung der Unit-Tests wurden lediglich die einfach zugänglichen Funktionen getestet. Um eine umfassendere Test-Abdeckung zu erreichen, können die Datenbank-Abfragen, sowie die Web-API-Anfragen ans GeoAdmin API gemockt werden.

## 7.2 Arbeiten an der cnlab App

- Im Rahmen der Analysephase wurde die cnlab App getestet. Die Verbesserungspunkte sind im Abschnitt 2.3.2 aufgeführt.

### **7.3 Arbeiten an der Auswertung zu Speedtest-Apps**

- Wie im Kapitel 2.2 „Auswertung zu Speedtest-Apps“ erwähnt, wird die Bewertung der erhobenen Daten zu den Speedtest-Apps durch die cnlab AG fertiggestellt, da den Studenten das Fachwissen dazu fehlt.

## Kapitel 8

# Schlussfolgerung

Im Rahmen dieser Bachelorarbeit wurden Netztests analysiert, Empfehlungen für Speedtest-Auswertungen erarbeitet und eine erweiterte Auswertung von Speedtest-Messungen implementiert.

Die genaue Betrachtung von bekannten Netztests ergab, dass diese mit sehr unterschiedlichen Messmethoden und Kriterien arbeiten. Aufgrund dieser Differenzen in der Erhebung und der Bewertung der Resultate gehen unterschiedliche Gewinner aus den Tests hervor. Somit ist es möglich, dass es für jeden Provider einen Netztest gibt, bei dem er sehr gut dargestellt wird und damit werben kann.

Bei den Statistiken wurden Schwankungen bei der Auswertung von Download-Datenraten untersucht. Dabei zeigte sich, dass die statistische Aussagekraft erhöht wird, wenn Messungen aus dem gesamten Zeitraum sowie unterschiedlichen Regionen vorhanden sind. Grund dafür ist, dass die Datenrate jedes Providers monatliche und regionale Unterschiede aufweist. Zudem wurde gezeigt, dass bei einer zufälligen Auswahl von Messungen keine Aussagen zu Minimal- und Maximalwerten gemacht werden können. Weiter wurden die Anzahl Werte bzw. der Anteil pro Provider bei einer zufälligen Auswahl erarbeitet, welche zu zuverlässigen Aussagen führen.

Mit dem Tagging-Server wurde eine automatische Kategorisierung von Speedtest-Messungen implementiert. Beim dafür verwendeten Algorithmus wurde komplett auf der „grünen Wiese“ angefangen. Trotzdem konnte bei den angenommenen Beispieldaten eine Korrektheit von über 80% erreicht werden.

Der Tagging-Server ist somit bereit, produktiv eingesetzt zu werden. Mit seiner Hilfe können differenziertere Auswertungen von Speedtest-Messungen erstellt werden.

Es gibt noch Fälle, wie zum Beispiel bei Messungen in oder neben einem Bahnhofsgebäude, bei denen eine falsche Kategorisierung vorgenommen wird. Diese Fälle gilt es nun genauer zu analysieren und Lösungsansätze auszuarbeiten.

## Kapitel 9

# Literaturverzeichnis

- [1] “GeoAdmin API 3.0 documentation.” [Online]. Available: <https://api3.geo.admin.ch/>
- [2] “OpenStreetMap Deutschland FAQs: Fragen und Antworten.” [Online]. Available: <https://www.openstreetmap.de/faq.html>
- [3] “Salt Mobile - Zuverlässige und schnelle Verbindung mit Salt - durch mehrere Tests bestätigt.” [Online]. Available: <https://www.salt.ch/de/test-results/>
- [4] “Unsere Auszeichnungen (Sunrise).” [Online]. Available: <https://www.sunrise.ch/de/privatkunden/sunrise-produkte/trustmarks.html>
- [5] “Tests und Auszeichnungen Swisscom Netz.” [Online]. Available: <https://www.swisscom.ch/de/about/unternehmen/portraet/netz/tests-und-auszeichnungen.html>
- [6] H. Rügheimer, “Mobilfunk-Netztest 2017: Schweizer Handy-Netze im Vergleich.” [Online]. Available: <http://www.connect.de/vergleich/netztest-2017-bestes-handy-netz-schweiz-mobilfunk-3196571.html>
- [7] —, “Mobilfunk-Netztest 2017: So testen connect und P3 communications das beste Handynetz.” [Online]. Available: <http://www.connect.de/ratgeber/mobilfunk-netztest-2017-so-testen-wir-3196575.html>
- [8] “Swisscom hauchdünn vor Sunrise - Der härteste Handy-Netztest der Schweiz: Swisscom, Salt und Sunrise im Test.” [Online]. Available: [http://www.chip.de/artikel/Der-haerteste-Handy-Netztest-der-Schweiz-Swisscom-Salt-und-Sunrise-im-Test-2\\_107564731.html](http://www.chip.de/artikel/Der-haerteste-Handy-Netztest-der-Schweiz-Swisscom-Salt-und-Sunrise-im-Test-2_107564731.html)
- [9] “Switzerland Fastest Broadband and Mobile Network Awards | Speedtest Awards.” [Online]. Available: <http://www.speedtest.net/awards/ch/carrier/2016>

- [10] M. Kowalsky, “Spannung im Netz,” *BILANZ*, no. 18/2016, pp. 44–47, Sep. 2016. [Online]. Available: [https://www.peopelfone.ch/documents/bilanz-telekom-rating-2016\\_de.pdf](https://www.peopelfone.ch/documents/bilanz-telekom-rating-2016_de.pdf)
- [11] “Mobilfunk-Netztest 2017: Ergebnisse - Endergebnis Schweiz.” [Online]. Available: <http://www.connect.de/bildergalerie/netztest-2017-mobilfunk-connect-ergebnisse-galerie-3196577-118653267.html>
- [12] “Switzerland Fastest Broadband and Mobile Network Awards | Speedtest Awards.” [Online]. Available: <http://www.speedtest.net/awards/ch/carrier/2016>
- [13] “M-Budget Mobile - Günstig telefonieren mit M-Budget.” [Online]. Available: <https://shop.m-budget.migros.ch/de/mobile-abonnement>
- [14] “Methodology | Speedtest Awards.” [Online]. Available: <http://www.speedtest.net/awards/methodology>
- [15] “Nah am Nutzer: So testen und bewerten wir - Der härteste Handy-Netztest der Schweiz: Swisscom, Salt und Sunrise im Test.” [Online]. Available: [http://www.chip.de/artikel/Der-haerteste-Handy-Netztest-der-Schweiz-Swisscom-Salt-und-Sunrise-im-Test-4\\_107564722.html](http://www.chip.de/artikel/Der-haerteste-Handy-Netztest-der-Schweiz-Swisscom-Salt-und-Sunrise-im-Test-4_107564722.html)
- [16] “Mobilfunk-Netztest 2017: Ergebnisse - Test-Route Schweiz.” [Online]. Available: <http://www.connect.de/bildergalerie/netztest-2017-mobilfunk-connect-ergebnisse-galerie-3196577-118653264.html>
- [17] cnlab AG, “cnlab Speed Test,” Apr. 2016. [Online]. Available: <https://play.google.com/store/apps/details?id=ch.cnlab.speedtest&hl=de>
- [18] “information technology research | cnlab.” [Online]. Available: <https://www.cnlab.ch/>
- [19] Ookla, “Speedtest.net,” Jan. 2017. [Online]. Available: <https://play.google.com/store/apps/details?id=org.zwanoo.android.speedtest&hl=de>
- [20] “Ookla | The world standard in Internet metrics.” [Online]. Available: <http://www.ookla.com/>
- [21] C. D. GmbH, “CHIP Netztest,” Jun. 2015. [Online]. Available: <https://play.google.com/store/apps/details?id=de.ncqa.floq.chip&hl=de>
- [22] “CHIP INFO.” [Online]. Available: <http://www.chip.info/>
- [23] OpenSignal.com, “Geschwindigkeits-Test 3g & 4g,” 2017. [Online]. Available: <https://play.google.com/store/apps/details?id=com.staircase3.opensignal&hl=de>
- [24] “3g and 4g LTE Cell Coverage Map - OpenSignal.” [Online]. Available: <https://opensignal.com/>

- [25] Sensorly.com, “4g map & speedtest,” Feb. 2017. [Online]. Available: <https://play.google.com/store/apps/details?id=com.sensorly.viewer&hl=de>
- [26] “Coverage Maps.” [Online]. Available: <http://www.sensorly.com/4g-cdma-gsm-wifi-coverage-maps-your-fingertips>
- [27] nPerf.com, “Speed Test & QoS 3g 4g WiFi,” 2016. [Online]. Available: <https://play.google.com/store/apps/details?id=com.nperf.tester&hl=de>
- [28] “nPerf speed test - Test your broadband connection - nPerf.” [Online]. Available: <https://www.nperf.com/en/>
- [29] R. GmbH, “Traffic Monitor+ mit Speedtest,” 2017. [Online]. Available: <https://play.google.com/store/apps/details?id=com.radioopt.templus&hl=de>
- [30] “Measure. Customer. Experience.” [Online]. Available: <https://www.radioopt.com/>
- [31] “High-Tech Gründerfonds | Vodafone übernimmt den Netzwerk-Analyse-Spezialisten RadioOpt.” [Online]. Available: <https://high-tech-gruenderfonds.de/de/vodafone-uebernimmt-den-netzwerk-analyse-spezialisten-radioopt/>
- [32] P. i. GmbH, “U get,” 2017. [Online]. Available: <https://play.google.com/store/apps/details?id=com.ugetapp&hl=de>
- [33] “P3 insight GmbH - subsidiary of P3 group.” [Online]. Available: <https://www.p3-group.com/en/about-us/executive-board/p3-insight-gmbh/>
- [34] “Germany, Austria, Switzerland 2017.” [Online]. Available: <http://www.connect-testmagazine.com/mobile-network-tests/germany-austria-switzerland-2017/>
- [35] C. S. Inc, “Cisco Data Meter,” Jul. 2016. [Online]. Available: <https://play.google.com/store/apps/details?id=com.cisco.android.cbt&hl=de>
- [36] “FLOQ Netztest (Speedtest) - Android-App.” [Online]. Available: <http://beste-apps.chip.de/android/app/floq-netztest-android-app,de.ncqa.floq/>
- [37] “FLOQ - Out of Order.” [Online]. Available: <https://www.floq.net/>
- [38] LIONMOBI, “Network Master - Speed Test,” 2017. [Online]. Available: <https://play.google.com/store/apps/details?id=com.lionmobi.netmaster&hl=de>
- [39] BeMobile, “Speed Test,” Jan. 2017. [Online]. Available: <https://play.google.com/store/apps/details?id=com.kbudev.speedtest&hl=de>
- [40] SpeedSpot.org, “Einfacher Speed Test,” Jan. 2017. [Online]. Available: <https://play.google.com/store/apps/details?id=org.speedspot.speedspotspeedtest&hl=de>

- [41] “Fastest Broadband and Mobile Network Awards | Speedtest Awards.” [Online]. Available: <http://www.speedtest.net/awards>
- [42] “404 Not Found.” [Online]. Available: <https://www.floq.net/karte>
- [43] “Terms & Conditions for mobile apps,” Dec. 2015. [Online]. Available: <https://termsfeed.com/blog/terms-conditions-mobile-apps/>
- [44] “Wi-Free | gratis im Internet surfen | UPC.” [Online]. Available: <https://www.upc.ch/de/internet/wi-free/>
- [45] “Coverage Maps.” [Online]. Available: [http://www.sensorly.com/map/4G/CH/Switzerland/Swisscom/lte\\_22801](http://www.sensorly.com/map/4G/CH/Switzerland/Swisscom/lte_22801)
- [46] “Ookla | Speedtest Intelligence from Ookla | Internet performance database.” [Online]. Available: <https://www.ookla.com/speedtest-intelligence>
- [47] “CHIP Netztest - Android-App - CHIP.” [Online]. Available: <http://beste-apps.chip.de/android/app/chip-netztest-android-app,de.ncqa.floq.chip/>
- [48] “Business Solutions - OpenSignal.” [Online]. Available: <https://opensignal.com/business-solutions>
- [49] “Business solutions.” [Online]. Available: <http://www.sensorly.com/business-solutions>
- [50] “About - nPerf.” [Online]. Available: <https://www.nperf.com/en/about-us/#privacyPolicy>
- [51] “cnlab Performance Benchmarking.” [Online]. Available: <https://ux.cnlab.ch/performance/overview/index.jsp>
- [52] “Aufgaben des ARE.” [Online]. Available: <https://www.are.admin.ch/are/de/home/are/aufgaben-des-bundesamts-fuer-raumentwicklung.html>
- [53] “Rechtliches.” [Online]. Available: <https://www.admin.ch/gov/de/start/rechtliches.html>
- [54] “Gemeindetypologie ARE, geocat.ch direct partners - swisstopo.” [Online]. Available: [https://www.geocat.ch/geonetwork/srv/ger/md.viewer#/full\\_view/c967f055-9482-4bcb-a1bb-8de7971e7293](https://www.geocat.ch/geonetwork/srv/ger/md.viewer#/full_view/c967f055-9482-4bcb-a1bb-8de7971e7293)
- [55] “API REST Services — GeoAdmin API 3.0 documentation.” [Online]. Available: <https://api3.geo.admin.ch/services/sdiservices.html>
- [56] “Schweizer Landeskoordinaten,” May 2017, page Version ID: 165605962. [Online]. Available: [https://de.wikipedia.org/w/index.php?title=Schweizer\\_Landeskoordinaten&oldid=165605962](https://de.wikipedia.org/w/index.php?title=Schweizer_Landeskoordinaten&oldid=165605962)
- [57] B. C, “node-postgres: PostgreSQL client for node.js,” Jun. 2017, original-date: 2010-10-15T23:05:50Z. [Online]. Available: <https://github.com/brianc/node-postgres>

- [58] “Leaflet — an open-source JavaScript library for interactive maps.” [Online]. Available: <http://leafletjs.com/>
- [59] European Telecommunications Standards Institute, “Digital cellular telecommunications system (Phase 2+); Background for Radio Frequency (RF) requirements (GSM 05.50 version 6.1.0 Release 1997),” Apr. 2000. [Online]. Available: [http://www.etsi.org/deliver/etsi\\_TR/101100\\_101199/101115/06.01.00\\_60/tr\\_101115v060100p.pdf](http://www.etsi.org/deliver/etsi_TR/101100_101199/101115/06.01.00_60/tr_101115v060100p.pdf)
- [60] S. Ahmadi, *LTE-advanced: a practical systems approach to understanding 3GPP LTE releases 10 and 11 radio access technologies*. Amsterdam: Elsevier, 2013.
- [61] “Die SBB fahren im Vergleich im Schnecken tempo,” *Handelszeitung*, Jan. 2014. [Online]. Available: <http://www.handelszeitung.ch/unternehmen/die-sbb-fahren-im-vergleich-im-schnecken-tempo-660389>
- [62] “Längengrad und Breitengrad – Erklärung.” [Online]. Available: <https://www.timeanddate.de/geographie/laengengrad-breitengrad>
- [63] Djexplo, “English: Illustration of geographic latitude and longitude of the earth,” May 2011. [Online]. Available: [https://commons.wikimedia.org/wiki/File:Latitude\\_and\\_Longitude\\_of\\_the\\_Earth.svg](https://commons.wikimedia.org/wiki/File:Latitude_and_Longitude_of_the_Earth.svg)
- [64] “Location | Android Developers.” [Online]. Available: [https://developer.android.com/reference/android/location/Location.html#getAccuracy\(\)](https://developer.android.com/reference/android/location/Location.html#getAccuracy())
- [65] “GPS Basics - learn.sparkfun.com.” [Online]. Available: <https://learn.sparkfun.com/tutorials/gps-basics>
- [66] R. S. King and M. Gröndahl, “How Google Collected Data From Wi-Fi Networks,” *The New York Times*, May 2012. [Online]. Available: [http://www.nytimes.com/interactive/2012/05/23/business/How-Google-Collected-Data-From-Wi-Fi-Networks.html?\\_r=0](http://www.nytimes.com/interactive/2012/05/23/business/How-Google-Collected-Data-From-Wi-Fi-Networks.html?_r=0)
- [67] “LocationManager | Android Developers.” [Online]. Available: [https://developer.android.com/reference/android/location/LocationManager.html#NETWORK\\_PROVIDER](https://developer.android.com/reference/android/location/LocationManager.html#NETWORK_PROVIDER)
- [68] “Ortung :: locating :: ITWissen.info.” [Online]. Available: <http://www.itwissen.info/Ortung-locating.html>
- [69] “Zugangspunkte in die Google-Standortdienste aufnehmen lassen - Nexus-Hilfe.” [Online]. Available: <https://support.google.com/nexus/answer/1725632?hl=de>
- [70] “Ortung und Positionsbestimmung mit Mobilfunk.” [Online]. Available: <https://www.elektronik-kompodium.de/sites/kom/1201061.htm>
- [71] “1-Mean Sea Level, GPS, and the Geoid.” [Online]. Available: <http://www.esri.com/news/arcuser/0703/geoid1of3.html>

- [72] “GpsPasSion Forums - GPS altitude : wgs84 / msl / geodetic corrections.” [Online]. Available: [http://www.gpspassion.com/forumsen/topic.asp?TOPIC\\_ID=10915](http://www.gpspassion.com/forumsen/topic.asp?TOPIC_ID=10915)
- [73] “JSON Schema.” [Online]. Available: <http://json-schema.org/>
- [74] “express-jsonschema.” [Online]. Available: <https://www.npmjs.com/package/express-jsonschema>
- [75] “ST\_distance.” [Online]. Available: [https://postgis.net/docs/ST\\_Distance.html](https://postgis.net/docs/ST_Distance.html)
- [76] “DE:Key:railway – OpenStreetMap Wiki.” [Online]. Available: <http://wiki.openstreetmap.org/wiki/DE:Key:railway>
- [77] “DE:Key:highway – OpenStreetMap Wiki.” [Online]. Available: <http://wiki.openstreetmap.org/wiki/DE:Key:highway>
- [78] “Key:footway – OpenStreetMap Wiki.” [Online]. Available: <http://wiki.openstreetmap.org/wiki/Key:footway>
- [79] “Gebäude,” Jun. 2017, page Version ID: 166198747. [Online]. Available: <https://de.wikipedia.org/wiki/Geb%C3%A4ude>
- [80] “DE:Key:tourism – OpenStreetMap Wiki.” [Online]. Available: <http://wiki.openstreetmap.org/wiki/DE:Key:tourism>
- [81] “Converting to WGS84 – OpenStreetMap Wiki.” [Online]. Available: [http://wiki.openstreetmap.org/wiki/Converting\\_to\\_WGS84](http://wiki.openstreetmap.org/wiki/Converting_to_WGS84)
- [82] “Education Center.” [Online]. Available: <http://boundlessgeo.com/resources/training/>
- [83] “Gemeindetypologie ARE GeoAdmin API.” [Online]. Available: <https://map.geo.admin.ch/?layers=ch.are.gemeindetypen>
- [84] “Raum mit städtischem Charakter 2012.” [Online]. Available: <https://www.bfs.admin.ch/bfs/de/home/statistiken/querschnittsthemen/raeumliche-analysen/raeumliche-gliederungen/analyseregionen.assetdetail.349558.html>
- [85] “Swiss Geoportal.” [Online]. Available: <https://map.geo.admin.ch>
- [86] “Bevölkerungsdichte GeoAdmin API.” [Online]. Available: <https://map.geo.admin.ch/?layers=ch.are.bevoelkerungsdichte>
- [87] “Key:boundary – OpenStreetMap Wiki.” [Online]. Available: <http://wiki.openstreetmap.org/wiki/Key:boundary>
- [88] “DE:Key:natural – OpenStreetMap Wiki.” [Online]. Available: <http://wiki.openstreetmap.org/wiki/DE:Key:natural>

- [89] “DE:Key:leisure – OpenStreetMap Wiki.” [Online]. Available: <http://wiki.openstreetmap.org/wiki/DE:Key:leisure>
- [90] “DE:Key:landuse – OpenStreetMap Wiki.” [Online]. Available: <http://wiki.openstreetmap.org/wiki/DE:Key:landuse>
- [91] “async.” [Online]. Available: <https://www.npmjs.com/package/async>
- [92] “async - Documentation.” [Online]. Available: <http://caolan.github.io/async/docs.html#parallel>
- [93] “Section 8: Spatial Indexing — Introduction to PostGIS.” [Online]. Available: <http://revenant.ca/www/postgis/workshop/indexing.html>
- [94] V. Schatz, “Usage of osm2pgsql.” [Online]. Available: <http://www.volkerschatz.com/net/osm/osm2pgsql-usage.html>
- [95] “PostgreSQL: Documentation: 9.5: VACUUM.” [Online]. Available: <https://www.postgresql.org/docs/9.5/static/sql-vacuum.html>
- [96] “DE:Key:maxspeed – OpenStreetMap Wiki.” [Online]. Available: <http://wiki.openstreetmap.org/wiki/DE:Key:maxspeed>
- [97] “Osm2pgsql – OpenStreetMap Wiki.” [Online]. Available: <http://wiki.openstreetmap.org/wiki/Osm2pgsql>
- [98] G. Williams, “[OSM-newbies] Howto: Query to get max speed limit of nearby road/highway,” Aug. 2012. [Online]. Available: <https://lists.openstreetmap.org/pipermail/newbies/2012-August/006997.html>
- [99] “postgis - osm2pgsql - can’t find ”maxspeed”tag after import - Geographic Information Systems Stack Exchange.” [Online]. Available: <https://gis.stackexchange.com/questions/135715/osm2pgsql-cant-find-maxspeed-tag-after-import>
- [100] “osm2po - openstreetmap converter and routing engine for java.” [Online]. Available: <http://osm2po.de/>
- [101] “Daten der Gemeindtypen GeoAdmin API.” [Online]. Available: <http://data.geo.admin.ch/ch.aren.gemeindtypen/>
- [102] “Neue Volkszählung, Bevölkerung, Privathaushalte: Geodaten.” [Online]. Available: <https://www.bfs.admin.ch/bfs/de/home/dienstleistungen/geostat/geodaten-bundesstatistik/gebäude-wohnungen-haushalte-personen/ergebnisse-volkszählung-ab-2010.assetdetail.1442443.html>
- [103] “GPS Almanach - Informationen über die Satelliten.” [Online]. Available: <http://www.quantenwelt.de/technik/GPS/almanach.html>

- [104] “Mock-Objekt,” May 2017, page Version ID: 165904611. [Online]. Available: <https://de.wikipedia.org/w/index.php?title=Mock-Objekt&oldid=165904611>
- [105] “geocat Home.” [Online]. Available: <https://www.geocat.admin.ch/de/home.html>
- [106] “Validator,” Jul. 2013, page Version ID: 120385769. [Online]. Available: <https://de.wikipedia.org/w/index.php?title=Validator&oldid=120385769>
- [107] “Was bedeutet WGS84 ?” [Online]. Available: [http://www.boerde-online.de/index.php?option=com\\_content&view=article&id=121:was-bedeutet-wgs84-&catid=93:faq&Itemid=69](http://www.boerde-online.de/index.php?option=com_content&view=article&id=121:was-bedeutet-wgs84-&catid=93:faq&Itemid=69)
- [108] “How do we import multipolygon relations from OpenStreetMap properly with osm2pgsql? - Geographic Information Systems Stack Exchange.” [Online]. Available: <https://gis.stackexchange.com/questions/228357/how-do-we-import-multipolygon-relations-from-openstreetmap-properly-with-osm2pgsql>
- [109] “How to update your OSM Tables with Changesets.” [Online]. Available: [http://rustyb.github.io/lesotho\\_manual/update/2015/02/27/update-osm-tables.html](http://rustyb.github.io/lesotho_manual/update/2015/02/27/update-osm-tables.html)
- [110] “npm.” [Online]. Available: <https://www.npmjs.com/>
- [111] “Node.js.” [Online]. Available: <https://nodejs.org/en/>
- [112] “express.” [Online]. Available: <https://www.npmjs.com/package/express>
- [113] “The MIT License | Open Source Initiative.” [Online]. Available: <https://opensource.org/licenses/MIT>
- [114] “request.” [Online]. Available: <https://www.npmjs.com/package/request>
- [115] “Apache License, Version 2.0 | Open Source Initiative.” [Online]. Available: <https://opensource.org/licenses/Apache-2.0>
- [116] “log.” [Online]. Available: <https://www.npmjs.com/package/log>
- [117] “Path | Node.js v8.1.0 Documentation.” [Online]. Available: <https://nodejs.org/docs/latest/api/path.html>
- [118] “node.js-Lizenz.” [Online]. Available: <https://raw.githubusercontent.com/nodejs/node/master/LICENSE>
- [119] “File System | Node.js v8.1.0 Documentation.” [Online]. Available: <https://nodejs.org/api/fs.html>
- [120] “chai.” [Online]. Available: <https://www.npmjs.com/package/chai>
- [121] “mocha.” [Online]. Available: <https://www.npmjs.com/package/mocha>

- [122] “rewire.” [Online]. Available: <https://www.npmjs.com/package/rewire>
- [123] “Artillery - a modern load testing toolkit.” [Online]. Available: <https://artillery.io/>
- [124] “Project OSRM.” [Online]. Available: <http://project-osrm.org/>
- [125] “barefoot: Java library for integrating the map into software and services with state-of-the-art online and offline map matching that can be used stand-alone and in the cloud,” Jun. 2017, original-date: 2014-06-05T13:19:58Z. [Online]. Available: <https://github.com/bmwcarit/barefoot>
- [126] “Mapbox.” [Online]. Available: <https://www.mapbox.com/>
- [127] “Mapbox API Documentation.” [Online]. Available: <https://www.mapbox.com/api-documentation/#introduction>
- [128] “map-matching: Map Matching based on GraphHopper i.e. ‘snap’ GPX records to the digital road network.” Jun. 2017, original-date: 2014-12-10T09:38:00Z. [Online]. Available: <https://github.com/graphhopper/map-matching>
- [129] “TrackMatching.” [Online]. Available: <https://mapmatching.3scale.net/>
- [130] “overpass turbo.” [Online]. Available: <https://overpass-turbo.eu/>
- [131] “Overpass API OpenStreetMap.” [Online]. Available: [http://wiki.openstreetmap.org/wiki/Overpass\\_API](http://wiki.openstreetmap.org/wiki/Overpass_API)
- [132] “Overpass API/Overpass QL – OpenStreetMap Wiki.” [Online]. Available: [http://wiki.openstreetmap.org/wiki/Overpass\\_API/Overpass\\_QL](http://wiki.openstreetmap.org/wiki/Overpass_API/Overpass_QL)
- [133] Frank, “docker-overpass-api: A fully working overpass-api docker image: cloning, replication, areas and www interface,” May 2017, original-date: 2016-11-06T16:11:41Z. [Online]. Available: <https://github.com/Frankkkkk/docker-overpass-api>

# Kapitel 10

## Abkürzungsverzeichnis (Glossar)

**AGB** Allgemeine Geschäftsbedingungen. 24, 25, 29

**Almanach** „Das Global Positioning System kann nur funktionieren, wenn man genaue Information hat, wo sich die Satelliten befinden und welche Zeit ihre internen Uhren zeigen. Diese Informationen werden in den Almanach-Daten gegeben, die jeder Satellit etwa alle 12,5 Minuten aussendet.“[103]. 62

**Bounding Box** Kleinst mögliches Rechteck, mit dem eine Geometrie umschlossen werden kann.. 81, 86

**FAQ** Frequently Asked Questions. 23

**gemockt** „Ein Mock-Objekt ist in der Softwareentwicklung ein Programmteil, das zur Durchführung von Modultests als Platzhalter für echte Objekte verwendet wird.“[104]. 127

**geocat** Bei geocat handelt es sich um den Metadatenkatalog für die Geodaten der Schweiz [105]. 49

**GPS** Global Positioning System. 62

**JSON** JavaScript Object Notation. 55, 59, 60, 64

**MAC** Media-Access-Control. 62

**npm** Node Package Manager. 56, 115

**OSM** Open Street Map. 55, 69, 74, 76, 80, 87, 88, 115

**RUP** Rational Unified Process. 106

**SBB** Schweizerische Bundesbahnen. 88

**Validator** „Ein Validator ist ein Computerprogramm, das die syntaktische Korrektheit eines Dokuments überprüft.“[106]. 64

**WGS 84** „Das World Geodetic System 1984 (WGS 84) ist ein [...] Referenzsystem als einheitliche Grundlage für Positionsangaben auf der Erde und im erdnahen Weltraum.“[107]. 54, 61, 63, 80

**WLAN** Wireless Local Area Network. 62

**Teil II**  
**Anhang**

# Kapitel 11

## Ablageorte

### 11.1 Arbeitsdokumente

Die für diese Arbeit erstellten weiterführenden Arbeitsdokumente sind unter dem Link <https://tinyurl.com/schlussabgabe> einsehbar. Der dort abgespeicherte Schlussabgabeordner wurde wie folgt aufgebaut:

- 1\_Analyse
  - 1\_Netzttests
  - 2\_Speedtest\_Apps
- 2\_Statistische\_Auswertungen
  - 0\_Archiv
  - 1\_Zufallsexperimente
  - 2\_Monatliche\_Auswertungen
  - 3\_Regionale\_Auswertungen
- 3\_Verwendete\_Datenquellen
- 4\_Software\_Engineering
- 5\_Realisierung
  - 0\_Archiv
  - 1\_Cnlab\_Daten
  - 2\_Testdata\_Server
  - 3\_Auswertung\_Manuelles\_Tagging
  - 4\_Diagramme

- 5\_Installation\_Auf\_Windows
- 6\_Installation\_Und\_Updates\_Auf\_Linux
- 6\_Projektmanagement
  - 1\_Aufgabenstellung
  - 2\_Zwischenpräsentation
  - 3\_Zeitplanung
  - 4\_Zeitauswertung
  - 5\_Sitzungsprotokolle
- 7\_Dokumentation
  - 1\_Abstract
  - 2\_Poster
  - 3\_EPrints\_Dokumentation

## 11.2 Code

Eine lauffähige Instanz des Tagging Server ist unter folgender URL erreichbar:  
<http://sinv-56041.edu.hsr.ch/api/v5.1/tag>

Die finale Version des Tagging-Server ist auf GitHub unter folgendem Link auffindbar:  
<https://github.com/dwin94/tagging-server-v5.1>

Den Tagging-Server mit all seinen Vorgängerversionen ist ebenfalls auf GitHub zu finden:  
<https://github.com/dwin94/tagging-server>

Der Testdata-Server, welcher für das manuelle Kategorisieren und das Auswerten benutzt wurde, ist wie folgt zu erreichen:  
<https://github.com/aha93/testdata-server>

Der ShowMeasurement-Server, welcher für die Anzeige einer einzelnen Messung anhand ihrer MeasurementID zuständig ist, kann unter folgendem Link bezogen werden:  
<https://github.com/aha93/showMeasurement>

Der erste umgesetzte Prototyp, bei dem es um den Einsatz von Overpass und Leaflet ging, ist wie folgt auf GitHub zu finden:  
<https://github.com/dwin94/BA-Task3-Prototype1>

# Kapitel 12

## Projektmanagementplan

### 12.1 Kostenvoranschlag

Das Projekt läuft im Rahmen der Bachelorarbeit. Diese sieht einen Personenaufwand von 360 Stunden pro Person vor, was bei einer 2-Personen-Gruppe einen Aufwand von 720 Stunden ergibt. Der Projektrahmen ist das Frühjahrssemester 2017, welches vom 20.02.2017 bis zum 16.06.2017 dauert und somit 17 Wochen umfasst.

### 12.2 Phasen / Iterationen

Das Projekt ist in die Phasen Inception, Elaboration, Construction und Transition, gemäss Rational Unified Process (RUP), aufgeteilt. Die Phasen sind, wie in der Grafik auf den nächsten Seiten ersichtlich, über das Frühjahrssemester verteilt. Ebenfalls ersichtlich sind die eingeplanten Meilensteine sowie die für weitere Fächer erwarteten Spitzenbelastung pro Woche.

#### **Rückblick zeitliche Planung**

Rückblickend gesehen kann gesagt werden, dass der erste aufgestellte Zeitplan gut erreicht werden konnte. Einzige Ausnahme stellte der Meilenstein „Statistiken abgeschlossen“ dar, dieser Meilenstein konnte erst ganz zum Schluss erledigt werden. Das lag daran, dass zweimal während dem Erstellen der Statistiken bemerkt wurde, dass die Filter für die validen Daten noch nicht stimmten.

# Zeitplan mit Spitzenbelastung

	Woche 1	Woche 2	Woche 3	Woche 4	Woche 5	Woche 6
Fach	20.02.-26.02.17	27.02.-05.03.17	06.03.-12.03.17	13.03.-19.03.17	20.03.-26.03.17	27.03.-02.04.17
HCID						
DBS2						
WED3						
<b>Arbeitsschritte bzw. Meilensteine</b>	Einarbeitung Tool-Setup	Umfeld-Analyse Einarbeitung App/Backend-Code Analyse Aufgabenstellung (Recherche)	Festlegung Aufgabenstellung Vertiefung in Punkte der Aufgabenstellung <b>Fertigstellung Analyse</b>	Statistik-Einarbeitung OpenStreetMap Einarbeitung	Vertiefung in Punkte der Aufgabenstellung Änderungen an Modellen <b>Statistiken abgeschlossen</b>	Vertiefung in Punkte der Aufgabenstellung
Iteration	Inception	Elaboration	Elaboration	Elaboration	Elaboration	Elaboration

## Farbschema:

normale Belastung	mittlere Belastung (Bsp. Bearbeitungszeit für Testate)	starke Belastung (Bsp. Testat-abgabe usw.)
-------------------	--	--

# Zeitplan mit Spitzenbelastung

	Woche 7	Woche 8	Woche 9	Woche 10	Woche 11	Woche 12
Fach	03.04.-09.04.17	10.04.-16.04.17	17.04.-23.04.17	24.04.-30.04.17	01.05.-07.05.17	08.05.-14.05.17
HCID						
DBS2						
WED3						

<b>Arbeitsschritte bzw. Meilensteine</b>	Funktionaler Prototyp <b>End of Elaboration</b>	Implementierung Punkt 1	Implementierung Punkt 1 <b>Version 1 implementiert</b>	Implementierung Punkt 2 Zwischenpräsentation	Implementierung Punkt 2 Usability-Tests	Implementierung Punkt 2
Iteration	Elaboration	Construction	Construction	Construction	Construction	Construction

## Farbschema:

normale Belastung	mittlere Belastung (Bsp. Bearbeitungszeit für Testate)	starke Belastung (Bsp. Testat-abgabe usw.)
-------------------	--	--

# Zeitplan mit Spitzenbelastung

	Woche 13	Woche 14	Woche 15	Woche 16	Woche 17
Fach	15.05.-21.05.17	22.05.-28.05.17	29.05.-04.06.17	05.06.-11.06.17	12.06.-18.06.17
HCID					
DBS2					
WED3					

<b>Arbeitsschritte bzw. Meilensteine</b>	Implementierung Punkt 3	Implementierung Punkt 3	Dokumentation <b>Implementation</b> abgeschlossen	Dokumentation Präsentation vorbereiten	Dokumentation abgeschlossen und Abgabe zusammengestellt
Iteration	Construction	Construction	Transition	Transition	Transition

## Farbschema:

normale Belastung	mittlere Belastung (Bsp. Bearbeitungszeit für Testate)	starke Belastung (Bsp. Testat-abgabe usw.)
-------------------	--	--

## 12.3 Meilensteine

Meilenstein	Woche soll	Woche ist	Beschreibung
Fertigstellung Analyse	Woche 3	Woche 3	Die Speedtest-App-Auswertung sowie die Netztestanalyse wurde fertiggestellt.
Statistiken abgeschlossen	Woche 5	Woche 17	Die statistischen Aussagen zu den zufälligen Messwerten wurden abgeschlossen.
End of Elaboration	Woche 7	Woche 7	Die Analyse und das Einarbeiten sind abgeschlossen und es ist bekannt, welche Arbeiten im Rahmen der Bachelorarbeiten angegangen werden.
Version 1 implementiert	Woche 9	Woche 9	Eine erste Version der Implementation wurde erstellt. Dies bedeutet, dass die OpenStreetMap-Datenbank eingerichtet wurde und der Server darauf zugreifen kann.
Implementation abgeschlossen	Woche 15	Woche 15	Alle Änderungen am Code wurden implementiert, sodass der Server einsatzfähig ist.
Schlussabgabe	Woche 17	Woche 17	Alle Dokumente wurden abgabekonform erstellt, die Dokumentation gebunden, der Code aufbereitet und alles an Herr Heinzmann abgegeben.

## 12.4 Zeiterfassung

Die Abbildungen 12.1 und 12.2 zeigen die Soll-Zeiten pro Woche gegenüber den effektiv auf die Arbeitspakete gebuchten Zeiten auf. Die zu Beginn der Bachelorarbeit zu wenig geleisteten Stunden wurden gegen Ende der Arbeit mehr als eingeholt. Es wurden schlussendlich, statt der 720 erwarteten Stunden, 834 Stunden geleistet. Es kann festgehalten werden, dass die Verteilung der Arbeitsstunden überaus fair war, denn Andrea Hauser leistete 413.25 Stunden und David Windler wand 421 Stunden auf.

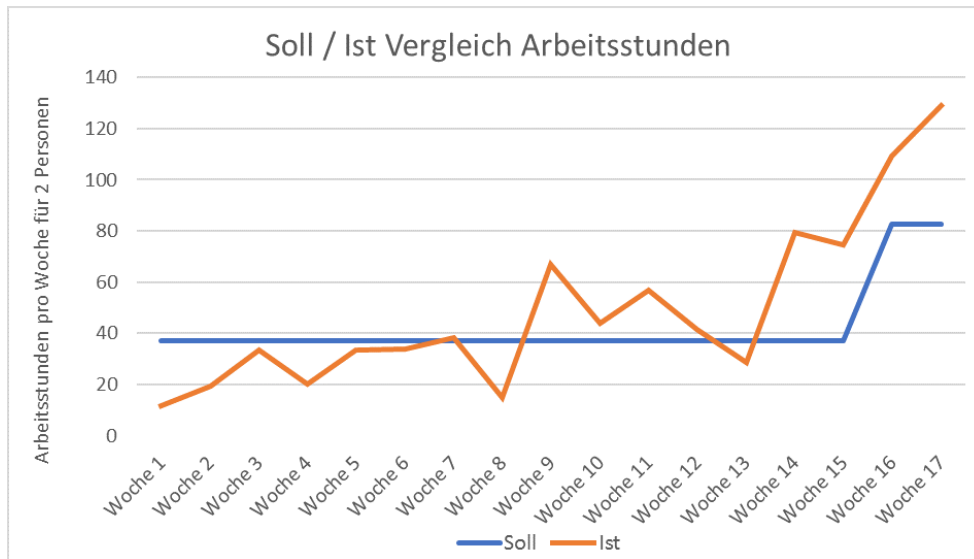


Abbildung 12.1: Darstellung des Soll / Ist Vergleichs der Arbeitsstunden

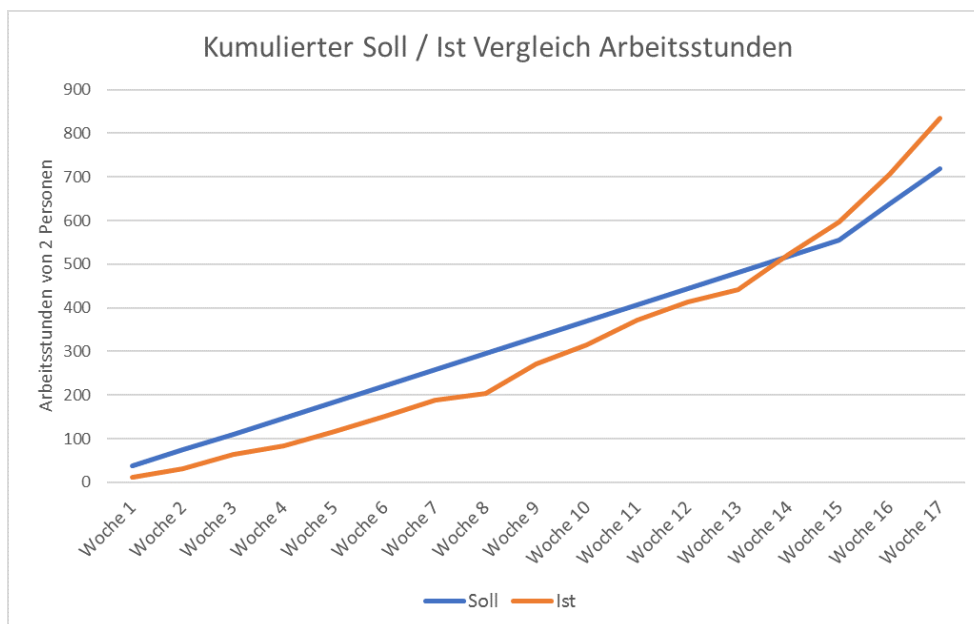


Abbildung 12.2: Darstellung der kumulierten Soll / Ist Arbeitsstunden

Auch in der Abbildung 12.3 ist erkennbar, dass gegen Ende der Arbeit mehr Zeit aufgewendet werden konnte. Dies zeigt sich besonders daran, dass die Transition-Phase, welche die letzten zwei Wochen umfasst, beinahe einen vierte der Arbeitszeit ausmacht.

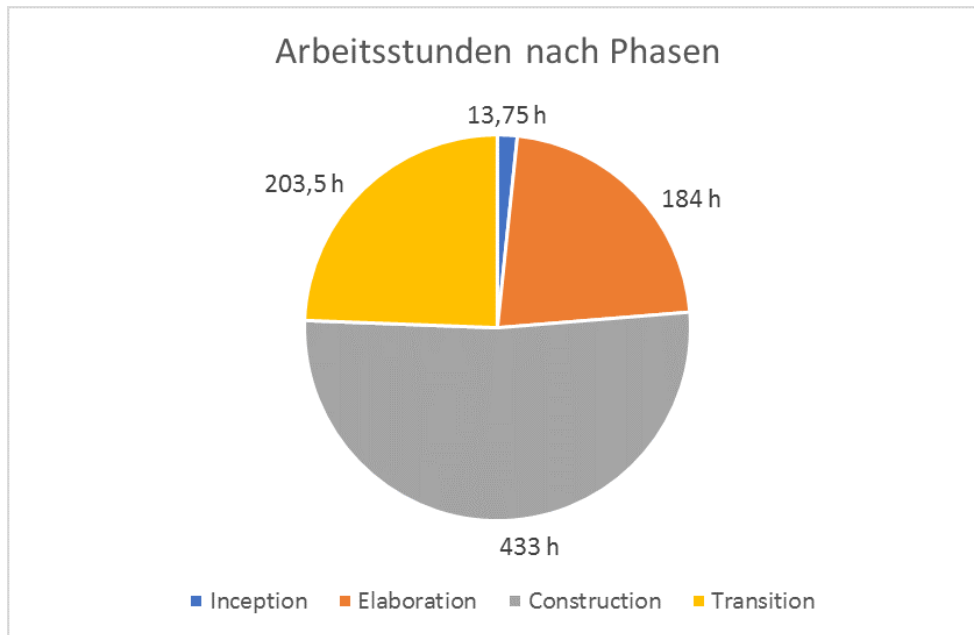


Abbildung 12.3: Darstellung der Arbeitsstunden nach den Phasen

Die unten stehende Abbildung 12.4 zeigt die Aufwände aufgeschlüsselt auf die einzelnen Kategorien. Mit einem Anteil von 35% der Gesamtzeit handelt es sich bei der Implementation um die grösste Kategorie.

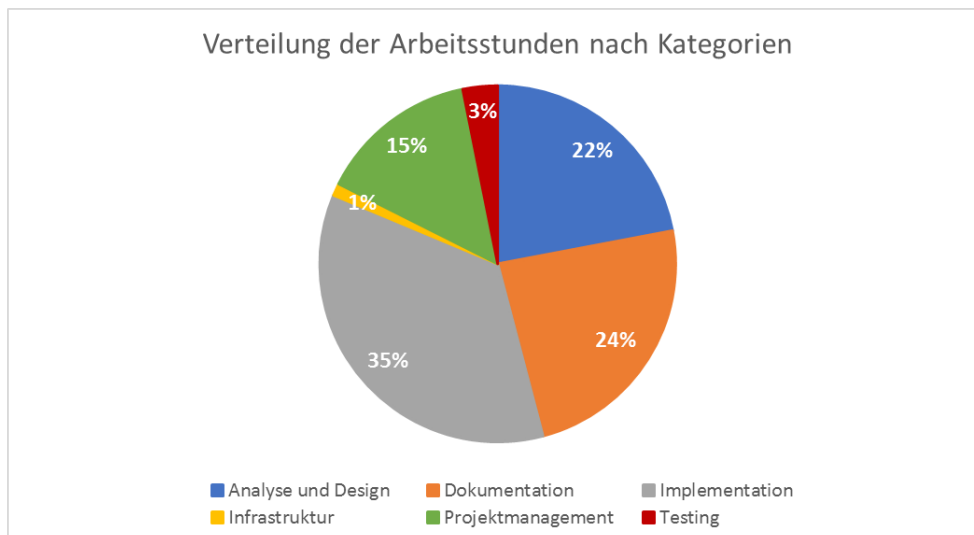


Abbildung 12.4: Darstellung der Arbeitsstunden nach den Kategorien

# Kapitel 13

## Risikomanagement

### 13.1 Umgang mit Risiken

Die Teammitglieder sind bereit, bei unerwarteten oder nicht vorhergesehenen Zwischenfällen das Arbeitspensum für die Bachelorarbeit zu erhöhen, um den fristgerechten Abschluss der Arbeit zu gewährleisten, solange es sich dabei nicht um einen Dauerzustand handelt. Zusätzlich wird bei der Schätzung der Aufwände immer darauf geachtet, für unerwartetes eine Reserve einzuplanen.

Die folgende Tabelle zeigt alle die technischen Risiken dieser Arbeit auf:

Risiko	Eintretenswahrscheinlichkeit	Vorbeugung	Verhalten beim Eintreten
Ausfall eines Laptops	gering	Regelmässige Commits des Codes und der Dokumentation	Herunterladen der zuletzt hochgeladenen Version
Antwortzeiten des Tagging-Servers sind zu gross	mittel	Abklärungen zu Performance-Optimierungen	Abklärung mit Herr Keller, ob alle Möglichkeiten ausgeschöpft wurden
Fehlerhafte Funktionen	gross	Unit-Testing	Fehler beheben
Anforderungen an Kategorisierung zu hoch	mittel	Früher Prototyp	Anforderungen mit Auftraggeber neu definieren

# Kapitel 14

## Details zur Lösungsfindung

### 14.1 Installation Tagging-Server

Für die Installation wurde ein Skript „setup\_tagging\_server.sh“ erstellt, welches folgendes ausführt:

- Aufsetzen der OpenStreetMap-Datenbank der Schweiz
- Einrichten des stündlichen Datenbank-Updates
- Einrichten und Starten des Tagging-Servers

Detaillierte Erklärungen zu den einzelnen Schritten sind im Skript selbst vorhanden. Dieses befindet sich mit den dazugehörigen Dateien auf GitHub unter folgendem Link:  
<https://github.com/dwin94/tagging-server/tree/master/setup>.

Vor dem Ausführen des Skriptes sind folgende Variablen in den Dateien „setup\_tagging\_server.sh“ und „update\_database.sh“ anzupassen:

- `WORK_DIRECTORY=/tagging/server`  
Hier werden alle vom Tagging-Server benötigten Dateien abgelegt.
- `DATABASE_NAME=switzerland`
- `DATABASE_PASSWORD=password`

Der Inhalt von „setup“ ist anschliessend in den Ordner `WORK_DIRECTORY` zu kopieren. Von dort wird das Skript wie folgt gestartet, damit die verwendeten Umgebungsvariablen korrekt exportiert werden.

```
1  chmod +x setup_tagging_server.sh
2  sed -i -e 's/\r$//' setup_tagging_server.sh
3  . ./setup_tagging_server.sh
```

Abbildung 14.1: Starten des Installations-Skriptes

**Beachten**

Die Open Street Map (OSM)-IDs von Relations werden beim Import durch osm2pgsql [97] negiert [108]. Aus diesem Grund wird bei der Prüfung, ob eine Koordinate in der Schweiz liegt, die OSM-ID -51701 verwendet, obwohl die Relation „Schweiz“ die ID 51701 aufweist.

**14.1.1 Updates**

Während den Updates werden Änderungsdateien, sogenannte „diff-files“, durch das Programm osm2pgsql [97] in die Datenbank eingespielt. Innerhalb dieser Dateien sind Änderungen der folgenden Art vorhanden [109]:

- Modification (Veränderung)  
Ein bestehender Eintrag wurde verändert.
- Deletion (Löschung)  
Ein bestehender Eintrag wurde gelöscht.
- Creation (Erstellung)  
Ein neuer Eintrag wurde hinzugefügt.

Der Inhalt einer Datei sieht somit wie folgt aus:

```
<modify>
  <node id="1077842297" version="2" timestamp="2015-02-17T17:50:23Z" uid="2311536"
  user="JordanKepler" changeset="28914810" lat="-29.8387748" lon="27.3556953"/>
</modify>
<delete>
  <node id="1234176303" version="1" timestamp="2015-02-18T02:01:30Z" uid="5616"
  user="Adrian Frith" changeset="7779859" lat="-29.3423239" lon="29.4524785"/>
</delete>
<create>
  <node id="3354821518" version="1" timestamp="2015-02-16T21:22:45Z" uid="2647326"
  user="teboho1702" changeset="28895890" lat="-29.2896602" lon="29.0574209"/>
</create>
```

Abbildung 14.2: Möglicher Datei-Inhalt einer OSM-Änderungsdatei [109]

**14.2 Verwendete Libraries / npm-Module**

Der Node Package Manager (npm)[110] ist der Paketmanager für Node.js [111]. Mit diesem können fast eine halbe Million JavaScript-Pakete bezogen werden. Der Grossteil davon steht frei zur Verfügung.

In diesem Projekt wurden unter anderem die folgenden Pakete verwendet:

<b>Name</b>	<b>Zweck</b>	<b>Einsatz im Projekt</b>	<b>Lizenz</b>
express [112]	Web-Framework für node.js	Erstellung des Servers und des Routings	MIT [113]
express-jsonschema [74]	Middleware zur JSON-Schema-Validation	Validation der Input-Daten	MIT
async [91]	Funktionen zum erleichterten Umgang mit asynchronem JavaScript	Parallele Abarbeitung bei der Berechnung der unterschiedlichen Tagging-Werte	MIT
request [114]	Erstellung von HTTP bzw. HTTPS-Anfragen	Anfragen ans GeoAdmin API	Apache-2.0 [115]
pg (node-postgres) [57]	PostgreSQL-Client für Node.js	Verbindung zur OSM-Datenbank	MIT
log [116]	Funktionen zur einfachen Erstellung von Log-Files	Erstellung der Datei error.log	MIT
path [117] (Teil von Node.js)	Funktionen zum erleichterten Umgang mit Datei- und Ordner-Pfaden	Erstellung des Log-Ordners	Lizenz von Node.js [118]
fs [119] (Teil von Node.js)	Funktionen zum erleichterten Umgang mit Dateien auf dem Computer	Lesen der Konfigurations-Dateien	Lizenz von Node.js
chai [120]	Funktionen zur einfachen Erstellung von Unit-Tests	Erstellen von Unit-Tests	MIT
mocha [121]	JavaScript Test-Framework	Erstellung von Unit-Tests	MIT
rewire [122]	Stellt spezielle Getter und Setter für JavaScript-Module zur Verfügung	Zugriff auf nicht exportierte Funktionen für die Unit-Tests	MIT

Für den Betrieb des Node.js-Servers sowie zur Auslieferung der Webseiten an den Client werden noch weitere Module verwendet, auf welche hier nicht detailliert eingegangen wird.

## 14.3 Auswertung der location-Korrektheit

### 14.3.1 Aufsetzen des Testing-Servers

Um die Korrektheit der location-Bestimmungen durch den Tagging-Server zu überprüfen, wurden über 5'000 Messungen der cnlab AG von Hand kategorisiert. Ein dafür entwickelter Server (Testdata-Server) schickte diese Messungen einzeln an den Tagging-Server, um anschliessend

das Resultat mit dem erwarteten Wert zu vergleichen. Der Testdata-Server kann unter folgender URL heruntergeladen werden:

<https://github.com/aha93/testdata-server>

Dieser verwendet eine MySQL-Datenbank, da die Messdaten der cnlab AG ebenfalls aus MySQL stammen. Um die Datenbank aufzusetzen, sind folgende SQL-Dateien im Abgabeordner unter „Realisierung\Testdata\_Server“ zu importieren:

- `mobile_data_log.sql`  
Beinhaltet alle Messdaten und deren manuell zugewiesene location.
- `resulting_correctness.sql`  
Speichert die berechnete Korrektheit pro Testdurchlauf.
- `returned_measurement_values.sql`  
Speichert die location sowie die Gewichtung der zurückgegebenen Werte des Tagging-Servers ab.

Schliesslich ist die Datei „config.json“ im Ordner „config“ abzulegen und mit den korrekten Daten zu befüllen.

### **14.3.2 Auswertung aufgrund des manuellen Taggings**

Wie bereits erwähnt, wurden über 5'000 Messungen der cnlab AG von Hand kategorisiert, um danach die Korrektheit des Tagging-Servers bestimmen zu können. In der Abbildung 14.3 sind die Resultate für die Version 5.1 des Tagging-Servers ersichtlich.

## Resultate des manuellen Taggings der Version v5.1

### Auswertung auf gesamte location

Anzahl Messungen	Anteil Richtige	Anteil Falsche	Anteil Unbekannte	Anteil Abgewiesene
4067 Messungen	46 %	9 %	2 %	43 %

### Auswertung der Abgewiesenen

Anzahl Messungen	Anteil Grösser 200 Meter Accuracy	Anteil Überall gleiche Zeit	Anteil Input-Positionen zu weit auseinander	Anteil Longitude oder Latitude 0
1769 Messungen	98 %	0 %	2 %	0 %

### Auswertung der Angenommenen

Anzahl Messungen	Anteil Richtige der Angenommenen	Anteil Falsche der Angenommenen	Anteil Unbekannte der Angenommenen
2298 Messungen	82 %	14 %	4 %

### Auswertung railway

Anzahl railway Messungen	Anteil Richtige der angenommenen railway	Anteil Falsche der angenommenen railway	Anteil Unbekannte der angenommenen railway	Anteil Abgewiesene railway
2431 Messungen	72 %	20 %	8 %	72 %

### Auswertung street

Anzahl street Messungen	Anteil Richtige der angenommenen street	Anteil Falsche der angenommenen street	Anteil Unbekannte der angenommenen street	Anteil Abgewiesene street
719 Messungen	92 %	5 %	3 %	1 %

### Auswertung building

Anzahl building Messungen	Anteil Richtige der angenommenen building	Anteil Falsche der angenommenen building	Anteil Unbekannte der angenommenen building	Anteil Abgewiesene building
917 Messungen	82 %	15 %	3 %	1 %

Abbildung 14.3: Anhand des manuellen Taggings berechnete Korrektheit des Testdata-Servers

Es ist darauf hinzuweisen, dass es sich bei den manuell kategorisierten Daten um ein spezielles Datenset handelt. Normalerweise werden Speedtest-Messungen zu Hause, in einem Gebäude, durchgeführt. In den von der cnlab AG erhaltenen Messungen sind deshalb tendenziell zu viele Zug- und Strassenmessungen und zu wenige Gebäudemessungen enthalten.

### 14.3.3 Verbesserungspunkte

Um herauszufinden, was zu den falsch getaggtten Werten führte, wurde ein weiterer Server (ShowMeasurement-Server) implementiert. Mit diesem Server kann eine einzelne Messung anhand ihrer MeasurementID betrachtet werden. Der Server holt sich anhand dieser Nummer die dazugehörigen Daten aus der Datenbank und fragt damit den Tagging-Server an. Dessen Resultat werden zurückgegeben und auf einer Karte dargestellt. Dies sieht folgendermassen aus:

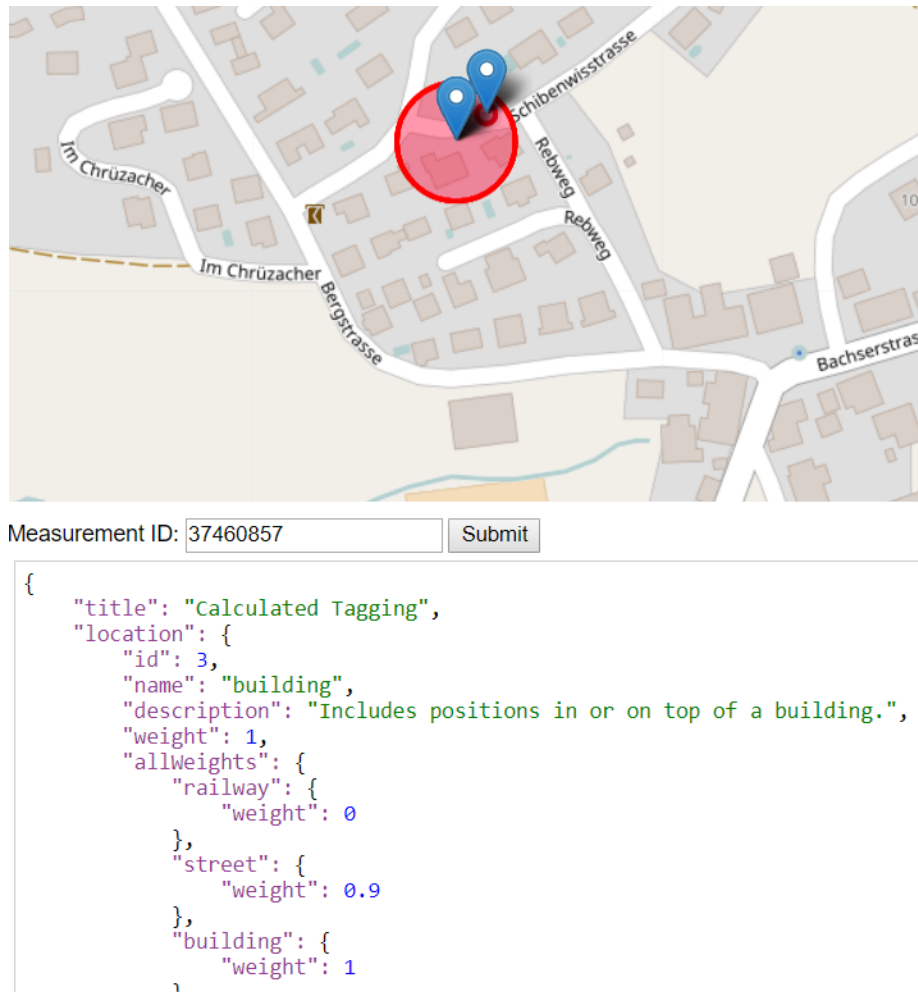


Abbildung 14.4: Ausschnitt aus der Ansicht des ShowMeasurement-Servers

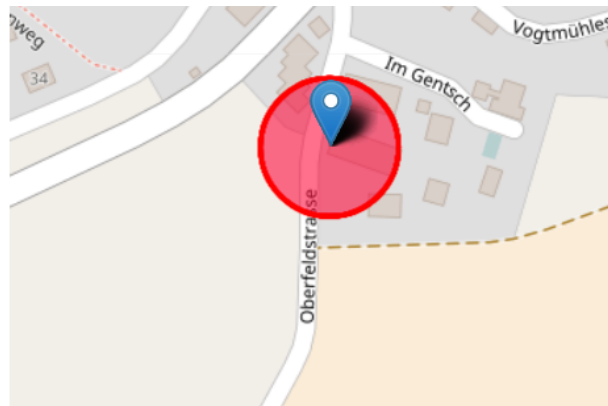
Dieser Server kann unter der URL <https://github.com/aha93/showMeasurement> bezogen werden. Er arbeitet mit der gleichen Datenbank wie der Testdata-Server und benötigt lediglich die bereits erwähnte Konfigurationsdatei.

Anhand des ShowMeasurement-Servers wurden die falsch getaggten Building- und Railway-Messungen analysiert, da diese die grössten Anteile an falsch getaggten Messungen aufwiesen. Es konnten folgende Probleme aufgedeckt werden:

**Messungen von Gebäuden** konnten nicht korrekt zugeordnet werden, wenn

- die gleiche Gewichtung mit anderen locations (v.a. street) vorlag.
  - Dies wurde zum Teil behoben, indem bei einer horizontalAccuracy von 20 bis 30 Metern die location building bevorzugt wird.

- Anzahl Messungen: ca. 190
- MeasurementID Beispielmessung: 37465668



Measurement ID:

```
{
  "title": "Calculated Tagging",
  "location": {
    "id": 2,
    "name": "street",
    "description": "Includes OpenStreetMap-",
    "weight": 1,
    "allWeights": {
      "railway": {
        "weight": 0
      },
      "street": {
        "weight": 1
      },
      "building": {
        "weight": 1
      }
    }
  }
}
```

Abbildung 14.5: Veranschaulichung der Problematik mit zwei gleichen Gewichtungen

- einzelne Punkte eine sehr grosse horizontalAccuracy (>1000 Meter) haben und zudem alle Messpunkte örtliche Sprünge aufweisen.
  - Diese Messungen konnten beim manuellen Tagging nur durch den Zusammenhang mit anderen Messungen zugeordnet werden.
  - Anzahl Messungen: ca. 15
  - MeasurementID Beispielmessung: 41110704

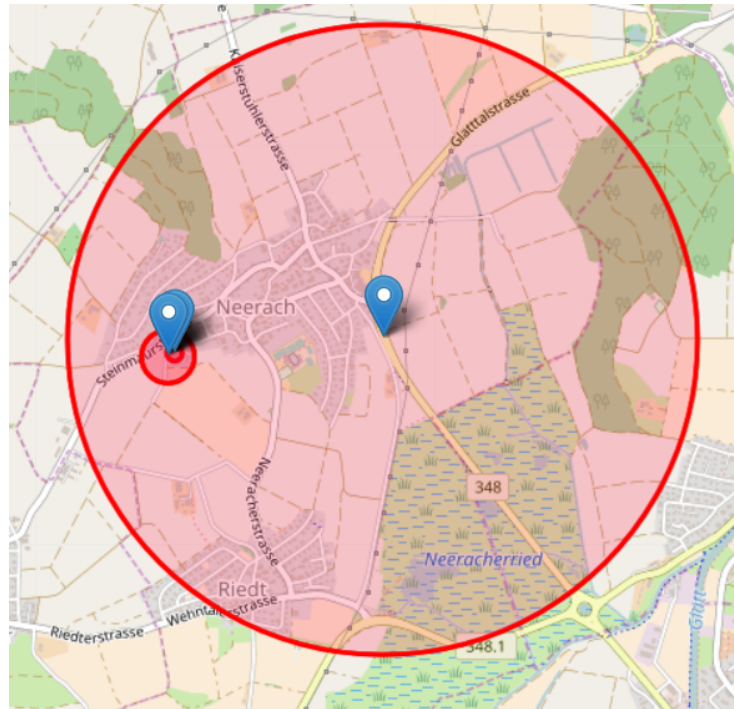


Abbildung 14.6: Veranschaulichung der Problematik mit grosser horizontalAccuracy

- sich das nächste Gebäude ausserhalb des Suchradius nach Gebäuden befand.
  - Dies konnte beim manuellen Tagging ebenfalls nur durch den Zusammenhang erkannt werden.
  - Anzahl Messungen: ca. 15
  - MeasurementID Beispielmessung: 37489383
- falsche Positionen mit einer tiefen horizontalAccuracy zurückgeliefert wurden.
  - Anzahl Messungen: 1
  - MeasurementID: 37507539

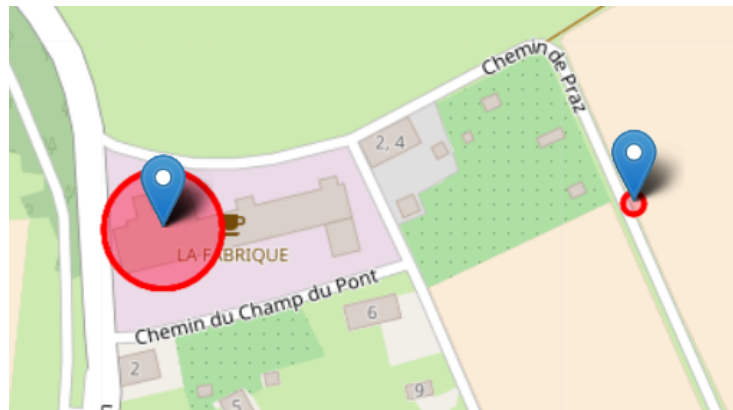
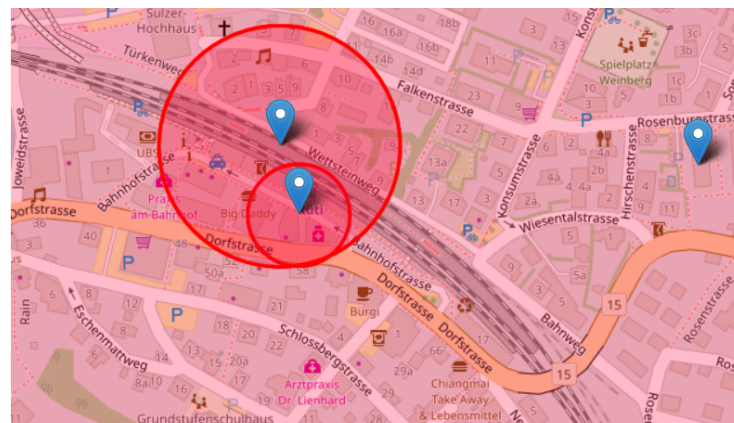


Abbildung 14.7: Veranschaulichung der Problematik mit tiefer horizontalAccuracy an falscher Stelle

- Gebäude, wie z.B. Hotels, nicht richtig in OSM eingetragen sind.
  - Dieses Problem wurde bereits behoben, indem neu auch mit dem OSM-Key „tourism“ nach Gebäuden gesucht wird.
- die Auflösung bei allen Punkten sehr ungenau ist. Als Folge ist die Geschwindigkeit jeweils zu gross.
  - Anzahl Messungen: ca. 3
  - MeasurementID Beispielmessung: 37508699

**Messungen von Zügen** konnten nicht korrekt zugeordnet werden, wenn

- sehr ungenaue Punkte die Messung verfälschen.
  - Häufig ist das Problem, dass bei den genaueren Punkten die Railways zu weit weg sind.
  - Anzahl Messungen: ca. 11
  - MeasurementID Beispielmessung: 39595133



Measurement ID:

```
{
  "title": "Calculated Tagging",
  "location": {
    "id": 2,
    "name": "street",
    "description": "Includes OpenStreetMap-Key: highway, Values: motorway, mo",
    "weight": 0.98,
    "allWeights": {
      "railway": {
        "weight": 0.29
      },
      "street": {
        "weight": 0.98
      }
    },
    "building": {
      "weight": 0
    }
  }
}
```

Abbildung 14.8: Veranschaulichung der Problematik, dass genauster Punkt zu weit weg von der Schiene ist

- die Auflösung der Messung sehr ungenau ist.
  - Dies konnte beim manuellen Tagging ebenfalls nur durch den Zusammenhang erkannt werden.
  - Anzahl Messungen: ca. 17
  - MeasurementID Beispielmessung: 41062805

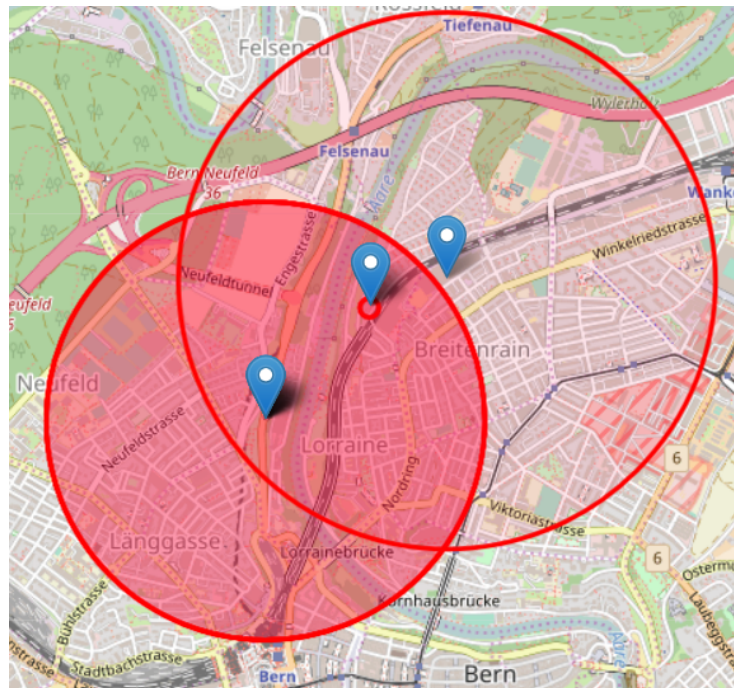


Abbildung 14.9: Veranschaulichung der Problematik von zu ungenauer horizontaler Auflösung

- die Zufahrt von Auge zwar klar erkennbar ist, aber der Algorithmus nur die einzelnen Punkte betrachtet.
  - Mögliche Lösung: einzelne Punkte der Messung verbinden und mit der damit erhaltenen Linie den am nächsten liegenden way suchen.
  - Anzahl Messungen: ca. 15
  - MeasurementID Beispielmessung: 37460276

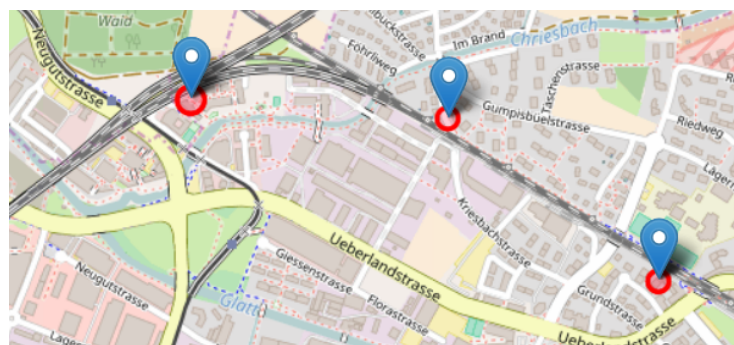


Abbildung 14.10: Von Auge genau zu erkennende Zugschiene welche aufgrund der unabhängigen Betrachtung der einzelnen Punkte nicht erkannt wird

- falsche Positionen mit einer tiefen horizontalAccuracy zurückgeliefert wurden.

- Anzahl Messungen: ca. 3
  - MeasurementID Beispielmessung: 37466498
- der Zug neben einem Bahnhofsgebäude steht und die horizontalAccuracy zwischen 20 und 30 Metern liegt.
  - Wegen der oben beschriebenen Umstellung, wird building hier stärker gewichtet als railway.
  - Anzahl Messungen: ca. 21
  - MeasurementID Beispielmessung: 39595382
- der Bahnsteig ein Dach hat.
  - Als mögliche Lösung könnte der Tag „building:roof“ ausgeschlossen werden. Es ist allerdings zu überprüfen, welche anderen Auswirkungen dies mit sich bringt.
  - Anzahl Messungen: ca. 4
  - MeasurementID Beispielmessung: 39602607
- die Positionen so ungenau aufgelöst wurden, dass alle zu weit weg von den Schienen sind.
  - Anzahl Messungen: ca. 61
  - MeasurementID Beispielmessung: 39654938
- eine Strasse direkt neben der Schiene liegt. Dieses Problem taucht meistens bei Tram-Messungen auf.
  - Anzahl Messungen: ca. 18
  - MeasurementID Beispielmessung: 41178321

## 14.4 Last-Tests

Um zu überprüfen, wie viele gleichzeitige Anfragen der Tagging-Server aushält und wie sich die Antwortzeiten verändern, wurden mit Artillery [123] Lasttests erstellt. Diese ergaben die folgenden Resultate:

- Szenario 1: Während 10 Sekunden 5 Anfragen pro Sekunde an den Server senden  
Resultat:
  - Minimale Antwortzeit: 264.5 ms
  - Maximale Antwortzeit: 1'173.2 ms
  - Median der Antwortzeiten: 302.7 ms
  - 95tes Perzentil der Antwortzeiten: 879.3 ms

- 99tes Perzentil der Antwortzeiten: 1'173.2 ms
- Szenario 2: Während 10 Sekunden 20 Anfragen pro Sekunde an den Server senden  
Resultat:
  - Minimale Antwortzeit: 220.1 ms
  - Maximale Antwortzeit: 1'236.1 ms
  - Median der Antwortzeiten: 282.9 ms
  - 95tes Perzentil der Antwortzeiten: 578.4 ms
  - 99tes Perzentil der Antwortzeiten: 713.2 ms
- Szenario 3: Während 10 Sekunden 30 Anfragen pro Sekunde an den Server senden  
Resultat:
  - Minimale Antwortzeit: 523.2 ms
  - Maximale Antwortzeit: 11'231.9 ms
  - Median der Antwortzeiten: 2'329.2 ms
  - 95tes Perzentil der Antwortzeiten: 5'171.8 ms
  - 99tes Perzentil der Antwortzeiten: 5'570 ms
- Szenario 4: Während 10 Sekunden 50 Anfragen pro Sekunde an den Server senden  
Resultat:
  - Minimale Antwortzeit: 820 ms
  - Maximale Antwortzeit: 25'844.7 ms
  - Median der Antwortzeiten: 9'303.9 ms
  - 95tes Perzentil der Antwortzeiten: 16'695.1 ms
  - 99tes Perzentil der Antwortzeiten: 17'844.5 ms
  - Von 500 Anfragen konnten 3 nicht abgeschlossen werden, da das GeoAdmin API [1] bei diesen keine Antwort gab.

Die Antwortzeit des Server beginnt sich also erst ab ca. 30 Anfragen pro Sekunden zu erhöhen. Gemäss Angaben der cnlab AG gibt es pro Tag ca. 1000 Speedtest-Messungen. Selbst wenn diese Messungen in nur einer Stunde des Tages beim Server eintreffen, so ergibt dies 0,28 Messungen pro Sekunde bzw. 1 Messung alle 3,6 Sekunden. Diese Last kann der Tagging-Server ohne Probleme aushalten.

## 14.5 Unit-Tests

Als Framework für die Erstellung der Unit-Test wurde Mocha [121] verwendet. Da Mocha nicht für die Überprüfung der getesteten Werte zuständig ist, musste zusätzlich eine assertion-Bibliothek, in unserem Fall chai [120], eingebunden werden. Die assertion-Bibliothek entscheidet, ob ein Test korrekt durchgelaufen ist oder nicht, in dem sie den erwarteten Wert mit dem erhaltenen Wert vergleicht. Da auch nicht exportierte Funktionen getestet wurden, wurde zusätzlich rewire [122] eingesetzt, welches den Zugriff auf solche Modul-internen Funktionen ermöglicht. Aufgrund der erstellten Unit-Test wurden vereinzelte Logikfehler gefunden und ansschliessend behoben.

Momentan wurden lediglich die Funktionen getestet, welche nicht auf die Datenbank oder auf ein externes Web-API zugreifen. In einem weiteren Schritt ist es sinnvoll, auch diese Funktionen zu testen, indem die Anfragen gemockt werden.

## 14.6 Produktabklärungen

Vor der Eigenentwicklung der Logik des Tagging-Servers wurden einige bereits vorhandene Tools im Bereich „Map Matching“ angeschaut. Dabei wird mittels aufgenommener GPS-Koordinaten versucht, die zurückgelegte Strecke wieder zu rekonstruieren. Die nachfolgend aufgeführten Tools wurden aus den erwähnten Gründen nicht verwendet.

- Open Source Routing Machine (OSRM) [124]  
Problem: Unterstützt keine Züge.
- Barefoot [125]  
Problem: Unterstützt keine Züge.
- Mapbox [126]  
Problem: Maximal 60 Anfragen pro Minute kostenlos [127]
- GraphHopper [128]  
Problem: Unterstützt keine Züge.
- TrackMatching [129]  
Problem: Nur für eine geringe Anfrage-Menge kostenlos.

## 14.7 Prototyp und Versionen

Die Idee, dass für das Tagging die Umgebung nach Schienen und Strassen abgefragt wird, entstand früh im Projekt. Auf der Suche nach einer Lösung wurde overpass turbo [130] entdeckt, eine Overpas-API-Instanz [131]. Diese stellt mittels einer eigenen Abfragesprache Queries [132] an OpenStreetMap (OSM).

Der erste erstellte Prototyp diente dazu, solche Abfragen ans Overpass API [131] zu stellen. Dazu wurde ein Bereich mitgegeben, in dem nach Zug-Strecken gesucht werden soll. Der Prototyp ist unter folgendem Link erreichbar: <https://github.com/dwin94/BA-Task3-Prototype1>

### BA-Task3-Prototype

Welcome to BA-Task3-Prototype

Query Overpass API

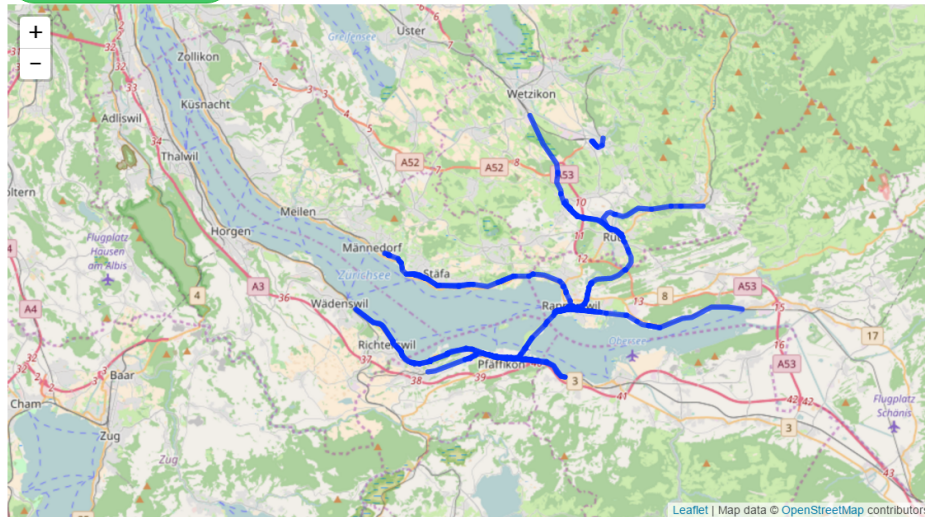


Abbildung 14.11: Erstellter Prototyp zum Abfragen von Zug-Strecken in einem definierten Bereich

Da die Overpass-Instanzen nicht mit zu vielen Anfragen belastet werden sollten, wurde zuerst versucht eine eigene Overpass-Docker-Instanz [133] zu erstellen. Das Aufsetzen schlug jedoch fehl. Dies führte dazu, dass der Umweg über das Overpass API auf OpenStreetMap nicht mehr gemacht wurde, sondern die OSM-Daten lokal in eine PostgreSQL-Datenbank geladen und direkt mittels SQL-Queries abgefragt wurden. Als Datenbank kam PostgreSQL zum Einsatz, da dies eine Empfehlung von Herr Keller war und es viele nützliche Tools rund um OSM und PostgreSQL gibt.

Die nächste Implementation war die erste Version des Tagging-Servers, welcher sich die Daten von der lokalen Datenbank holte. Es folgten sechs Weiterentwicklungen des Servers, bei denen die Kategorisierung der Speedtest-Messungen Stück um Stück verbessert und erweitert wurde. Der Code mit allen Versionen ist unter folgender URL erreichbar: <https://github.com/dwin94/tagging-server>

## 14.8 Lizenzabklärungen

## 14.8.1 Lizenzabklärung OSM

### Hauser Andrea

---

**Von:** Keller Stefan  
**Gesendet:** Dienstag, 28. März 2017 21:31  
**An:** Hauser Andrea; Windler David  
**Cc:** Heinzmann Peter Extern [peter.heinzmann@cnlab.ch]  
**Betreff:** AW: Use Cases Mobile Performance Mapping;  
OpenStreetMap und Bakom

Liebe Frau Hauser, lieber Herr Windler

Ich habe (gute) Neuigkeiten für Ihre Arbeit.

Kurz: Bei der Lizenzfrage liegt Ihre Anwendung eher auf der „erlaubten Seite“, so dass OSM genutzt werden kann, wobei die Frage sich erst durch ein (oder zwei ☺) Juristen abschliessend klären liesse.

Bei der Lösung Ihres sog. Map Matching-Problems habe ich einige Ideen – sowohl zu Algorithmen (\*) als auch zur OSM-Datenaufbereitung.

Haben Sie morgen 14:00 Uhr Zeit?

LG, S. Keller

(\*) <http://lists.openstreetmap.ch/pipermail/talk-ch/2017-March/004027.html>

---

**Von:** Hauser Andrea  
**Gesendet:** Mittwoch, 22. März 2017 16:46  
**An:** Keller Stefan  
**Cc:** Windler David  
**Betreff:** Use Cases Mobile Performance Mapping; OpenStreetMap und Bakom

Sehr geehrter Herr Keller

Wie gestern bei der Sitzung besprochen, senden wir Ihnen untenstehend die genaue Definition der Use Cases.

Use Case 1:

Es wird ein Bewegungsprofil der Messung festgelegt. Es soll also bestimmt werden können, ob sich die Person die am Messen ist im Zug befindet oder mit dem Auto unterwegs ist usw. Aus dieser

berechneten Information wird ein Flag in der Datenbank von cnlab entstehen. Die Angaben ob sich eine Strasse oder ein Gleis usw. in der Nähe der Messung befindet, soll von OpenStreetMap kommen. Die Standortdaten bzw. GPS-Daten kommen von cnlab bzw. den Personen, die mit der App von cnlab Messungen vornehmen.

Use Case 2:

Anhand von Antennenstandorten soll ausgewertet und angezeigt werden, wie viele Antennen es entlang einer Bahnstrecke (z.B. Zürich – Bern) hat. Es wird auch berechnet, wie weit die Antennen von der Bahnstrecke entfernt sind. Dadurch soll eine Aussage zur möglichen Verbindungsqualität des Handynetzes gemacht werden können. Die Daten zu den Zugstrecken soll von OpenStreetMap kommen, die Daten zu den Antennenstandorten wollen wir vom Bakom beziehen. Auf folgender Website <http://api3.geo.admin.ch/api/faq/index.html#which-layers-are-available> sind die dafür benötigten layers angegeben, es handelt sich um:

- 142 [ch.bakom.mobil-antennenstandorte-gsm](#) (Antennenstandorte GSM)
- 143 [ch.bakom.mobil-antennenstandorte-lte](#) (Antennenstandorte 4G (LTE))
- 144 [ch.bakom.mobil-antennenstandorte-umts](#) (Antennenstandorte 3G (UMTS))

Use Case 3:

Der Benutzer soll die Möglichkeit haben, die Antennenstandorte um eine Bestimmte Koordinate herum anzuzeigen. Er kann sich somit zum Beispiel anzeigen, welche Antennen es um seinen Wohnort herum hat oder welche Antennen sich um einen bestimmten Messpunkt herum befinden. Dabei soll die Karte von OpenStreetMap kommen und die Antennen wären wieder die von Bakom, wie im Use Case 2 bereits genauer erläutert.

Besten Dank für Ihre Abklärungen und freundliche Grüsse  
Andrea Hauser und David Windler

## 14.8.2 Lizenzabklärung ARE

### Hauser Andrea

---

**Von:** rolf.giezendanner@are.admin.ch  
**Gesendet:** Dienstag, 23. Mai 2017 15:31  
**An:** Hauser Andrea  
**Betreff:** AW: Lizenzfrage Gemeindetypen

Liebe Frau Hauser

Ja, aus unserer Sicht ist die Verwendung der Daten in Ordnung.

Freundliche Grüsse  
Rolf Giezendanner

---

**Von:** Hauser Andrea [mailto:andrea.hauser@hsr.ch]  
**Gesendet:** Dienstag, 23. Mai 2017 15:16  
**An:** Giezendanner Rolf ARE <rolf.giezendanner@are.admin.ch>  
**Betreff:** WG: Lizenzfrage Gemeindetypen

Sehr geehrter Herr Giezendanner

Wie von Ihnen empfohlen habe ich das Ganze mit Swisstopo abgeklärt. Aufgrund der Antwort von Swisstopo wollte ich nochmals sicher gehen, dass die Verwendung der Daten meinerseits in Ordnung ist. Können Sie mir bestätigen, dass dies der Fall ist?

Wie bereits geschildert, wird ein bestehender Datensatz um die Attribute typ\_code, kt\_kz, kt\_no, label und bfs\_no vom Gemeindetyp sowie um popd\_ha von Bevölkerungsdichte erweitert. Falls Sie noch weitere Angaben benötigen, um dies genauer zu beurteilen, stehe ich Ihnen gerne zu Verfügung.

Besten Dank für Ihre Antwort und freundliche Grüsse  
Andrea Hauser

---

**Von:** Kühni Christian swisstopo [mailto:Christian.Kuehni@swisstopo.ch]  
**Gesendet:** Freitag, 19. Mai 2017 14:14  
**An:** Hauser Andrea <andrea.hauser@hsr.ch>  
**Cc:** Giezendanner Rolf ARE <rolf.giezendanner@are.admin.ch>  
**Betreff:** AW: Lizenzfrage Gemeindetypen

Sehr geehrte Frau Hauser

Ich entschuldige mich bei Ihnen für die späte Antwort.

Die Nutzung der Daten muss grundsätzlich mit dem [Disclaimer](#) der Bundesverwaltung vereinbar sein. Aus dem Disclaimer dürfte besonders folgender Abschnitt für Sie relevant sein. „Für die Reproduktion jeglicher Elemente ist die schriftliche Zustimmung der Urheberrechtsträger im Voraus einzuholen.“

Bei dem von Ihnen verwendeten Datensatz handelt es sich um einen Datensatz des ARE und unserer Einschätzung nach obliegt die Erteilung der Zustimmung zur Nutzung beim entsprechenden Datenherrn.

Ich hoffe ich konnte Ihnen damit weiterhelfen und stehe Ihnen bei weiteren Fragen gerne zur Verfügung.

Freundliche Grüsse

---

Christian Kühni  
Geodatenabgabe

Eidgenössisches Departement für Verteidigung,  
Bevölkerungsschutz und Sport VBS  
Bundesamt für Landestopografie swisstopo

Seftigenstrasse 264, CH-3084 Wabern  
+41 58 469 05 28 (Tel. direkt)  
+41 58 469 04 59 (Fax)  
+41 58 469 01 11 (Tel. Zentrale)  
[Mailto:christian.kuehni@swisstopo.ch](mailto:christian.kuehni@swisstopo.ch)  
[www.swisstopo.ch](http://www.swisstopo.ch)  
Erreichbarkeit: Mo, Di & Fr

**Folgen Sie swisstopo!**

[Facebook](#) / [Twitter](#) / [Twitter \(Französisch\)](#) / [Newsletter](#)

---

**Von:** Hauser Andrea [<mailto:andrea.hauser@hsr.ch>]

**Gesendet:** Freitag, 12. Mai 2017 15:24

**An:** \_swisstopo-Geodata <[Geodata@swisstopo.ch](mailto:Geodata@swisstopo.ch)>

**Betreff:** WG: Lizenzfrage Gemeindetypen

Sehr geehrte Damen und Herren

Im Rahmen meiner Bachelorarbeit möchte ich gerne das Identify-Feature des GeoAdmin APIs für die folgenden Layer benutzen: ch.are.gemeindetypen sowie ch.are.bevoelkerungsdichte.

Leider konnte ich auf Ihrer Website nur Lizenz-Informationen im Zusammenhang mit zurückgelieferten Karten / Pixeln finden. In der Arbeit werden aber keine Karten benötigt, sondern nur das zurückgelieferte JSON-Objekt. Damit wird ein bestehender Datensatz um die Attribute typ\_code, kt\_kz, kt\_no, label und bfs\_no vom Gemeindetyp sowie um popd\_ha von Bevölkerungsdichte erweitert.

Beispiel einer Abfrage:

<https://api3.geo.admin.ch/rest/services/all/MapServer/identify?geometry=715116.7,209171.7&geometryFormat=geojson&geometryType=esriGeometryPoint&imageDisplay=1,1,1&lang=de&layers=all:ch.aren.gemeindetypen&mapExtent=0,0,1,1&returnGeometry=false&tolerance=5>

Was muss ich im Rahmen meiner Bachelorarbeit im Zusammenhang mit den Lizenzen beachten? Und was muss weiter beachtet werden, falls die Arbeit nach Abschluss in einer Firma kommerziell eingesetzt werden soll?

Besten Dank für Ihre Antwort und freundliche Grüsse  
Andrea Hauser

---

**Von:** [rolf.giezendanner@are.admin.ch](mailto:rolf.giezendanner@are.admin.ch) [<mailto:rolf.giezendanner@are.admin.ch>]

**Gesendet:** Freitag, 12. Mai 2017 15:12

**An:** Hauser Andrea <[andrea.hauser@hsr.ch](mailto:andrea.hauser@hsr.ch)>

**Betreff:** AW: Lizenzfrage Gemeindetypen

Guten Tag Frau Hauser

Vielen Dank für Ihre Anfrage.

Die Nutzungsbestimmungen der Geodaten kommen in Ihrem Fall nicht zur Anwendung. Für die Nutzung des API ist das Bundesamt für Landestopografie swisstopo zuständig. Die entsprechenden Bestimmungen finden Sie hier:

<https://www.geo.admin.ch/de/geo-dienstleistungen/geodienste/darstellungsdienste-webmapping-webgis-anwendungen/programmierschnittstelle-api/anmeldung.html>

Falls sich daraus Unklarheiten ergeben, bitte ich Sie, direkt mit swisstopo Kontakt aufzunehmen.

Viel Erfolg bei Ihrer Arbeit und freundliche Grüsse  
Rolf Giezendanner

**Rolf Giezendanner**

Leiter GIS-Fachstelle / Stv. Sektionschef

Eidgenössisches Departement für Umwelt, Verkehr, Energie und Kommunikation UVEK  
Bundesamt für Raumentwicklung ARE  
Sektion Grundlagen

Worbentalstrasse 66, CH-3063 Ittigen, Postadresse: CH-3003 Bern  
Tel +41 58 462 01 43  
[rolf.giezendanner@are.admin.ch](mailto:rolf.giezendanner@are.admin.ch)  
[www.are.admin.ch](http://www.are.admin.ch)

---

**Von:** Hauser Andrea [<mailto:andrea.hauser@hsr.ch>]  
**Gesendet:** Freitag, 12. Mai 2017 14:58  
**An:** Giezendanner Rolf ARE <[rolf.giezendanner@are.admin.ch](mailto:rolf.giezendanner@are.admin.ch)>  
**Betreff:** Lizenzfrage Gemeindetypen

Sehr geehrter Herr Giezendanner

Im Rahmen meiner Bachelorarbeit möchte ich gerne das Identify-Feature des GeoAdmin APIs für den folgenden Layer benutzen: ch.are.gemeindetypen.

Leider konnte ich auf Ihrer Website nur Lizenz-Informationen im Zusammenhang mit zurückgelieferten Karten / Pixeln finden. In der Arbeit werden aber keine Karten benötigt, sondern nur das zurückgelieferte JSON-Objekt. Damit wird ein bestehender Datensatz um die Attribute typ\_code, kt\_kz, kt\_no, label und bfs\_no erweitert.

Beispiel einer Abfrage:

<https://api3.geo.admin.ch/rest/services/all/MapServer/identify?geometry=715116.7,209171.7&geometryFormat=geojson&geometryType=esriGeometryPoint&imageDisplay=1,1,1&lang=de&layers=all:ch.are.gemeindetypen&mapExtent=0,0,1,1&returnGeometry=false&tolerance=5>

Was muss ich im Rahmen meiner Bachelorarbeit im Zusammenhang mit den Lizenzen beachten? Und was muss weiter beachtet werden, falls die Arbeit nach Abschluss in einer Firma kommerziell eingesetzt werden soll?

Besten Dank für Ihre Antwort und freundliche Grüsse  
Andrea Hauser

## **Kapitel 15**

# **Plakat zur Bachelorarbeit**

Das Plakat zur Bachelorarbeit ist auf der nächsten Seite eingebunden.

# Mobile Performance (Implementation neuer statistischer Auswertungsmöglichkeiten und Überprüfung der Aussagekraft von Netztest-Kampagnen)

Bachelorarbeit Frühjahrssemester 2017  
 Internet-Technologien und -Anwendungen



Andrea Hauser



David Windler

Betreuer: Prof. Dr. Peter Heinzmann

Experte: Dr. Thomas Siegenthaler

Projektpartner: HSR, Rapperswil SG

## Wozu braucht es eine erweiterte Auswertung?

Die cnlab AG bietet mit ihrer Speedtest-App die Möglichkeit, die Mobile Internetgeschwindigkeit zu messen. Die Messdaten werden ausgewertet und immer wieder von Medien bei Diskussionen zu Internet Qualitätsparametern beigezogen.

Um mittelfristig einen zuverlässigeren Vergleich von mobilen Netzanbietern zu ermöglichen, werden die in dieser Arbeit umgesetzten Klassierungen benötigt.

## Ziele

- Überprüfung der Aussagekraft von bekannten Netztests
- Untersuchung, unter welchen Bedingungen Speedtest-Ergebnisse statistisch aussagekräftig sind
- Erweiterte Auswertungsmöglichkeiten von Mess-Resultaten der cnlab Speedtest-App

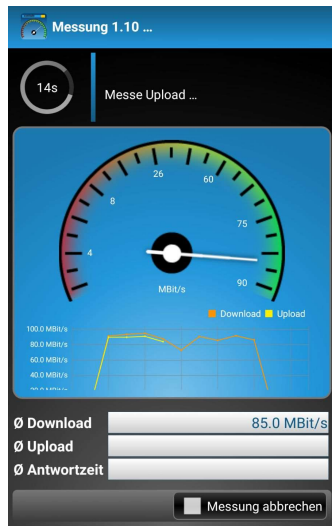
## Vorgehen

- Bekannte Netztests analysiert
- Diverse Mobile-Speedtest-Applikationen untersucht
- Algorithmus zur erweiterten Auswertung realisiert

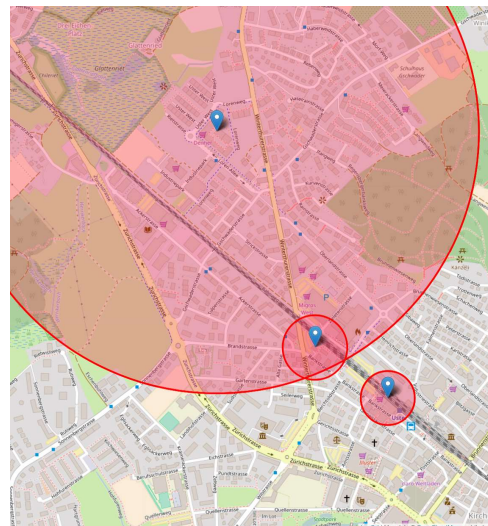
## Ergebnis

- Messmethoden der Netztests weichen so voneinander ab, dass jeder Provider einmal gewinnt.
- Bei Gegenüberstellung des 50ten Perzentils aller Messungen eines Providers zu den 50ten Perzentilen der einzelnen Monate treten Abweichungen von bis zu -40,53% auf.
- Algorithmus, welcher Speedtest-Messungen nach folgenden Kriterien klassiert:
  - Messort (Schiene, Strasse, Gebäude)
  - Fortbewegungsart
  - Bevölkerungsdichte und Gemeindetyp
  - Geografische Umgebung

137 von 142



Messung der Speedtest-App



Mess-Koordinaten mit Genauigkeitsradius

# Kapitel 16

## Verwendete Werkzeuge

### 16.1 Dokumentenverwaltung

**OneDrive** ist ein Dienst von Microsoft, um Dateien auf einem zentralen Speicherort abzulegen. Auf diesen kann über das Internet zugegriffen werden.

- Einsatzzweck: Dokumentenablage
- Version: 17.3
- Bezugsquelle: <https://onedrive.live.com/about/de-de/download/>
- Beachten: Benötigt kostenlose Registrierung auf <https://onedrive.live.com/>

**GitHub** ist ein Online-Versionsverwaltungssystem für Software.

- Einsatzzweck: Versionskontrolle für Code und Latex-Projektdokumentation
- Version: unbekannt
- Webseite: <https://github.com/>
- Beachten: Benötigt kostenlose Registrierung auf <https://github.com/join>. Gratis Private-Repositories gibt es als Studenten mit der Registrierung auf <https://education.github.com/pack>

**GitHub Desktop** ist ein Programm für Windows und macOS, um GitHub-Repositories zu lokal synchronisieren und verwalten.

- Einsatzzweck: Synchronisation von Code und Latex-Projektdokumentation
- Version: 3.3.4.0
- Bezugsquelle: <https://desktop.github.com/>

## 16.2 Server-Zugriff

**FileZilla Client** ist ein Programm für Windows, macOS und Linux, um mittels FTP (File Transfer Protocol) Daten auf einen Server hoch- und herunterzuladen.

- Einsatzzweck: Dateien auf HSR-Server hochladen
- Version: 3.26.1
- Bezugsquelle: <http://filezilla.de/>

**PuTTY** ist ein Programm für Windows und Linux, um Verbindungen mittels SSH (Secure Shell), Telnet oder über eine serielle Schnittstelle herzustellen.

- Einsatzzweck: SSH-Verbindung zum HSR-Server, um Installationen oder Konfigurationen vorzunehmen.
- Version: 0.67
- Bezugsquelle: <http://www.putty.org/>

## 16.3 Projektverwaltung

**Jira** ist eine webbasierte Projektmanagement-Software.

- Einsatzzweck: Projektplanung, Ticketverwaltung und Zeiterfassung
- Version: unbekannt
- Bezugsquelle: <https://de.atlassian.com/software/jira>

## 16.4 Dokumentation

**Microsoft Office** ist ein Paket von Büro-Software für Windows, macOS und weitere Betriebssysteme.

- Einsatzzweck: Erstellung von Sitzungsprotokollen Excel-Tabellen zur statistischen Auswertung
- Version: 1704
- Bezugsquelle: <https://products.office.com/>
- Beachten: Das Office-Paket kann als HSR-Student kostenlos heruntergeladen werden.

**TeXstudio** ist ein LaTeX-Editor für Windows, macOS und Linux.

- Einsatzzweck: Erstellung von LaTeX-Dokumenten, vor allem für Projektdokumentation
- Version: 2.12.4
- Bezugsquelle: <http://www.texstudio.org/>
- Beachten: Für die Erstellung von LaTeX-Dokumenten benötigt es eine TeX-Distribution (siehe MiKTeX). Weiter ist ein Perl-Interpreter Voraussetzung, um ein Glossar zu erstellen.

**MiKTeX** ist eine TeX-Distribution für Windows.

- Einsatzzweck: Interpretation und Kompilation von LaTeX-Dokumenten
- Version: 2.9
- Bezugsquelle: <https://miktex.org/download>

**ActivePerl** ist ein Perl-Interpreter für Windows.

- Einsatzzweck: Erstellung von LaTeX-Glossaren
- Version: 5.24.0
- Bezugsquelle: <http://www.activestate.com/activeperl>

**Zotero** ist eine Quellenverwaltungs-Software für Windows, macOS und Linux.

- Einsatzzweck: Quellenverwaltung
- Version: 4.0.29.17
- Bezugsquelle: <https://www.zotero.org/download/>
- Beachten: Für das Speichern von neuen Quellen eignet sich das Browser-AddOn. Weiter können die Exporteinstellungen des Zotero-Standalone auf BibTeX eingestellt werden, was es ermöglicht, neue Quellen per Drag und Drop in eine .bib-Datei einzufügen. So können neue Quellen schnell und einfach in LaTeX eingebunden werden.

## 16.5 Software-Entwicklung

**JetBrains WebStorm** ist eine JavaScript-Entwicklungsumgebung für Windows, macOS und Linux.

- Einsatzzweck: Entwicklung der Webserver
- Version: 27.1.4
- Bezugsquelle: <https://www.jetbrains.com/webstorm/>
- Beachten: Ist für Studenten und Lehrer mit Registrierung unter <https://www.jetbrains.com/student/> gratis.

**Node.js** ist eine Laufzeitumgebung für JavaScript.

- Einsatzzweck: Ausführung der Webserver
- Version: 7.10.0
- Bezugsquelle: <https://nodejs.org/en/>
- Beachten: Für Node.js wird der Node Package Manager benötigt (siehe unten).

**Node Package Manager (npm)** ist ein Paketmanager für Node.js.

- Einsatzzweck: Bezug der benötigten Pakete für die Webserver.
- Version: 5.0.3
- Bezugsquelle: <https://www.npmjs.com/get-npm>
- Beachten: Für Node.js wird der Node Package Manager (npm) benötigt (siehe unten)

**Postman** ist ein Tool mit dem Web-APIs getestet werden können.

- Einsatzzweck: Manuelles überprüfen des entwickelten Webserver
- Version: 5.0.0
- Bezugsquelle: <https://www.getpostman.com/apps>

**Google Chrome** ist ein Webbrowser.

- Einsatzzweck: Beobachtung von Netzwerkverkehrs zum und vom Webserver
- Version: 59.0.3071.86
- Bezugsquelle: <https://www.google.de/chrome/browser/desktop/index.html>

**XAMPP Control Panel** ist ein Programmpaket für Windows, macOS und Linux. Es enthält Apache, MariaDB, PHP und Perl.

- Einsatzzweck: Aufsetzen der Datenbank für das manuelle Tagging
- Version: 5.6.24
- Bezugsquelle: <https://www.apachefriends.org/de/index.html>

**PostgreSQL** ist ein objekt-relationales Datenbanksystem.

- Einsatzzweck: Lokaler Import und Zugriff auf OpenStreetMap-Daten
- Version: 9.6
- Bezugsquelle: <https://www.postgresql.org/download/>

**PostGIS** ist eine räumliche Datenbankeerweiterung für PostgreSQL.

- Einsatzzweck: Nutzung von Funktionen für den Umgang mit räumlichen Daten
- Version: 2.3
- Bezugsquelle: <http://postgis.net/install/>

**pgAdmin** ist eine Verwaltungs- und Entwicklungsplattform für PostgreSQL.

- Einsatzzweck: Zusammenstellen von SQL-Queries
- Version: pgAdmin 4 v1.5
- Bezugsquelle: <https://www.pgadmin.org/download/>

**osm2pgsql** ist ein Tool um OpenStreetMap-Daten in eine PostgreSQL-Datenbank laden zu können.

- Einsatzzweck: Laden der OSM-Daten der Schweiz in PostgreSQL
- Version: 0.90.2
- Bezugsquelle: <https://github.com/openstreetmap/osm2pgsql>

**osmosis** ist ein Tool um OpenStreetMap-Daten verarbeiten zu können.

- Einsatzzweck: Erstellung der Update-Dateien von OpenStreetMap
- Version: 0.45
- Bezugsquelle: <https://github.com/openstreetmap/osmosis>