

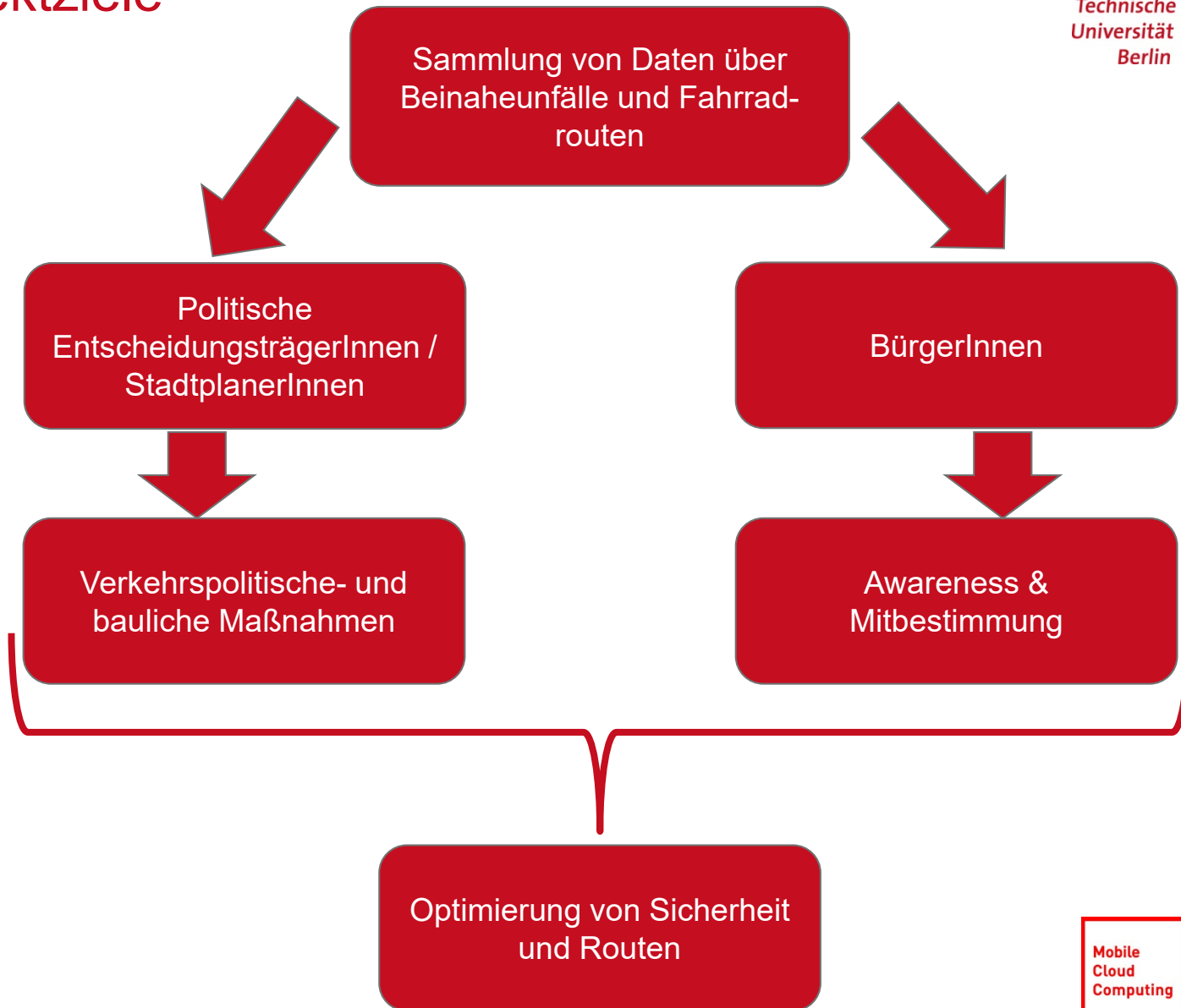


© D. Bermbach

SimRa: Sicherheit im Radverkehr

David Bermbach

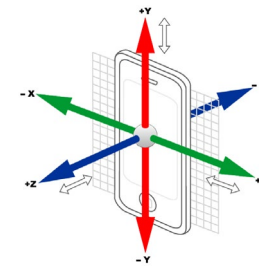
Projektziele



Hauptidee

Beschleunigungssensoren für die Erkennung von plötzlichen Bewegungen

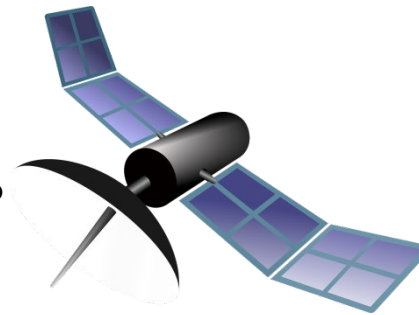
- Möglicher (Beinahe-)Unfall



<https://www.researchgate.net/profile/Jair-Garcia-Junior/publication/307754785/figure/fig/1/AS:459827492200448@1486642825367/figure-a2-Representacao-dos-eixos-do-acelerometro.png>

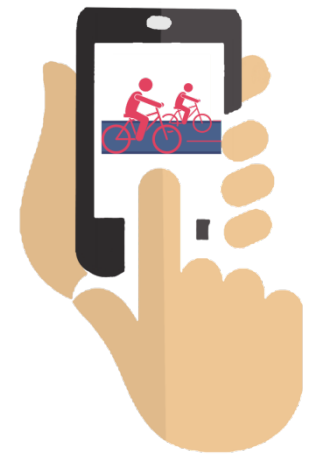
GPS-Koordinaten:

- Viel befahrene Straßenabschnitte
- Wo häufen sich (Beinahe-)Unfälle?

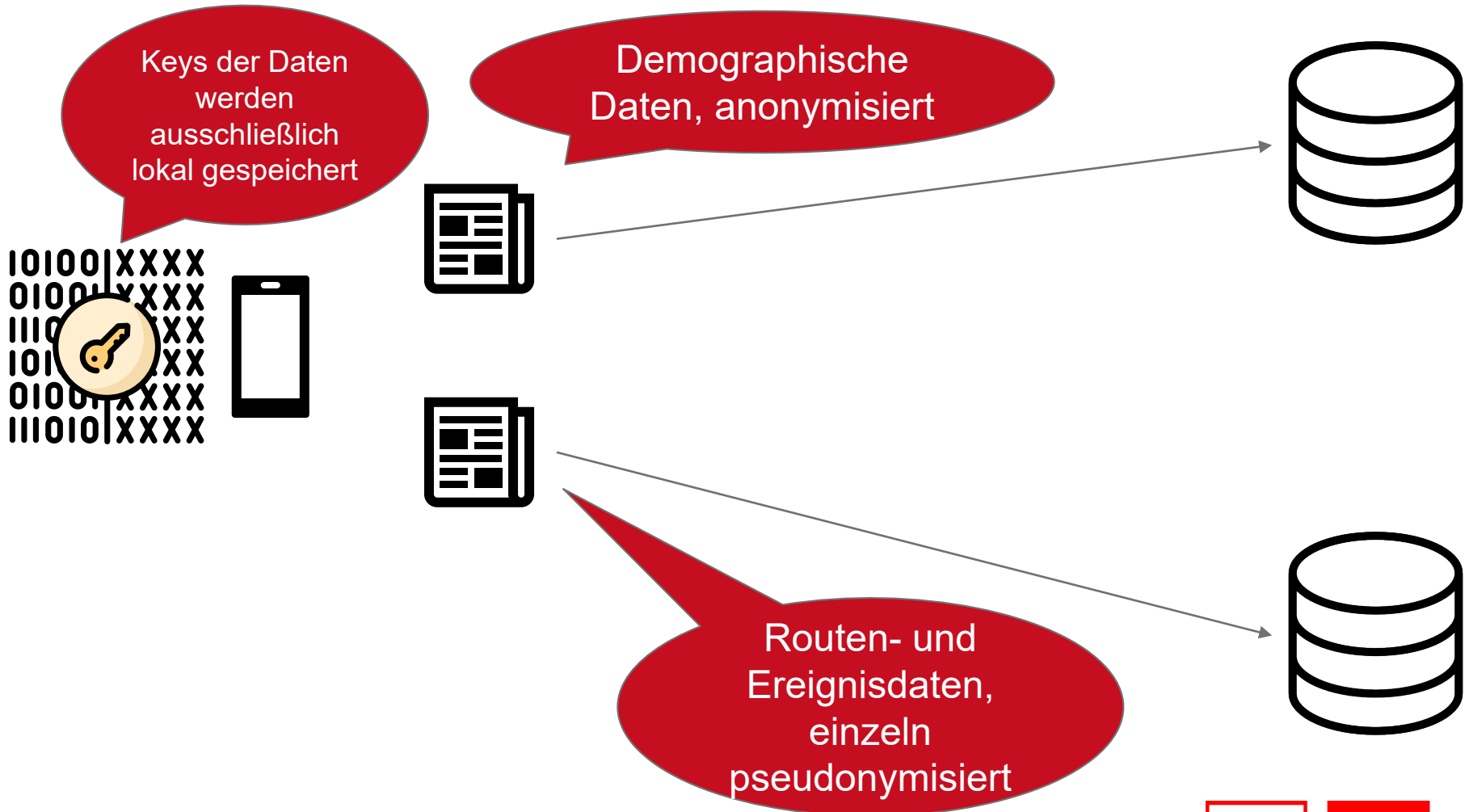


Ergänzung durch Nutzer

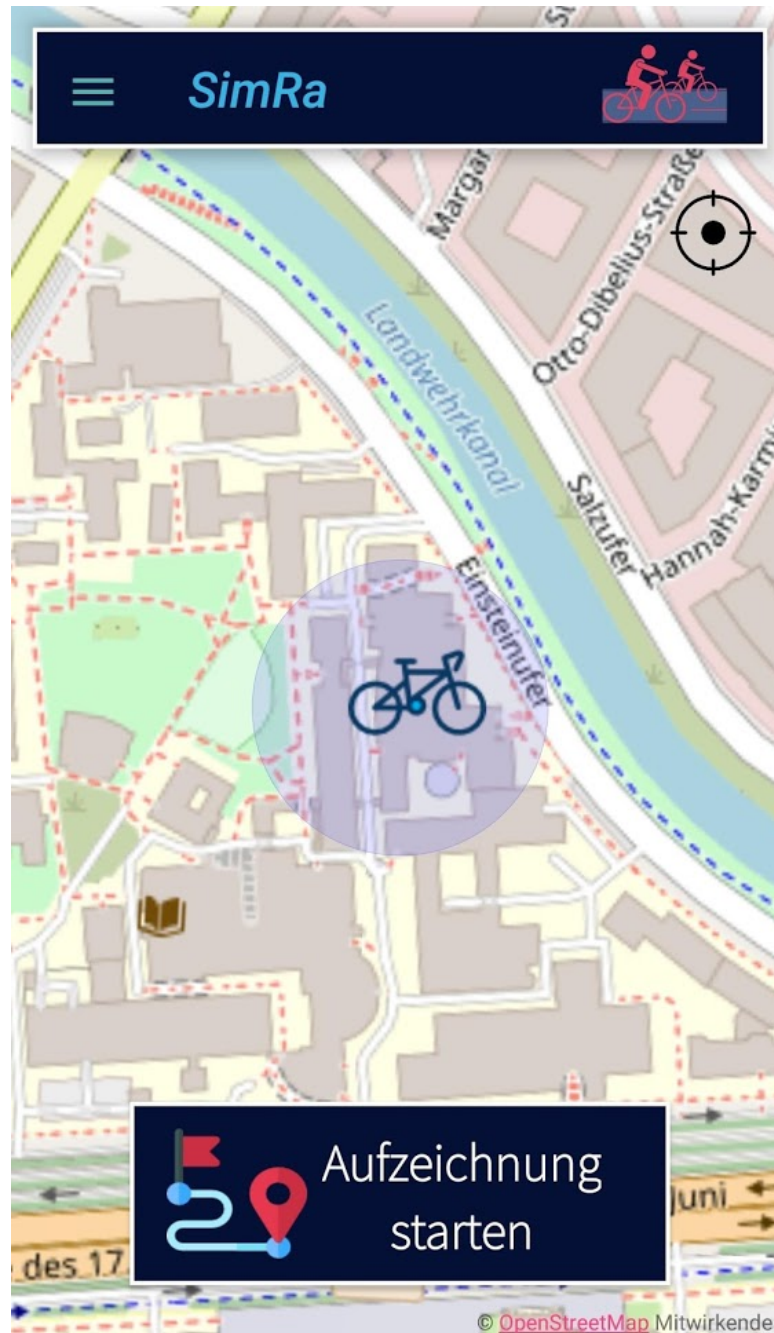
- Der Nutzer annotiert die Ereignisse



Fokus auf Privacy



SIMRA APP



Fahrteinstellungen

Unten finden Sie die Einstellungen, die für diese Fahrt verwendet werden. Sie können diese sowohl in diesem Fenster, als auch unter "Einstellungen" im Hauptmenü finden.

Fahrradtyp

Renntag

Üblicher Ort des Geräts während der Fahrt:

Fahrradkorb/Satteltasche

Auf dem Fahrrad wird ein Kind transportiert.

Am Fahrrad ist ein Anhänger angebracht.

Diese Einstellungen merken

FERTIG





Art des Ereignisses
Beinahe-Abbiegeunfall

Beteiligte Verkehrsteilnehmer

PKW Fußgänger
 Taxi Fahrrad
 Lieferwagen Motorrad
 Bus Sonstiges
 LKW (Bitte unten
angeben)

Diese Erfahrung war beängstigend

Optionaler Kommentar
Mein Kommentar

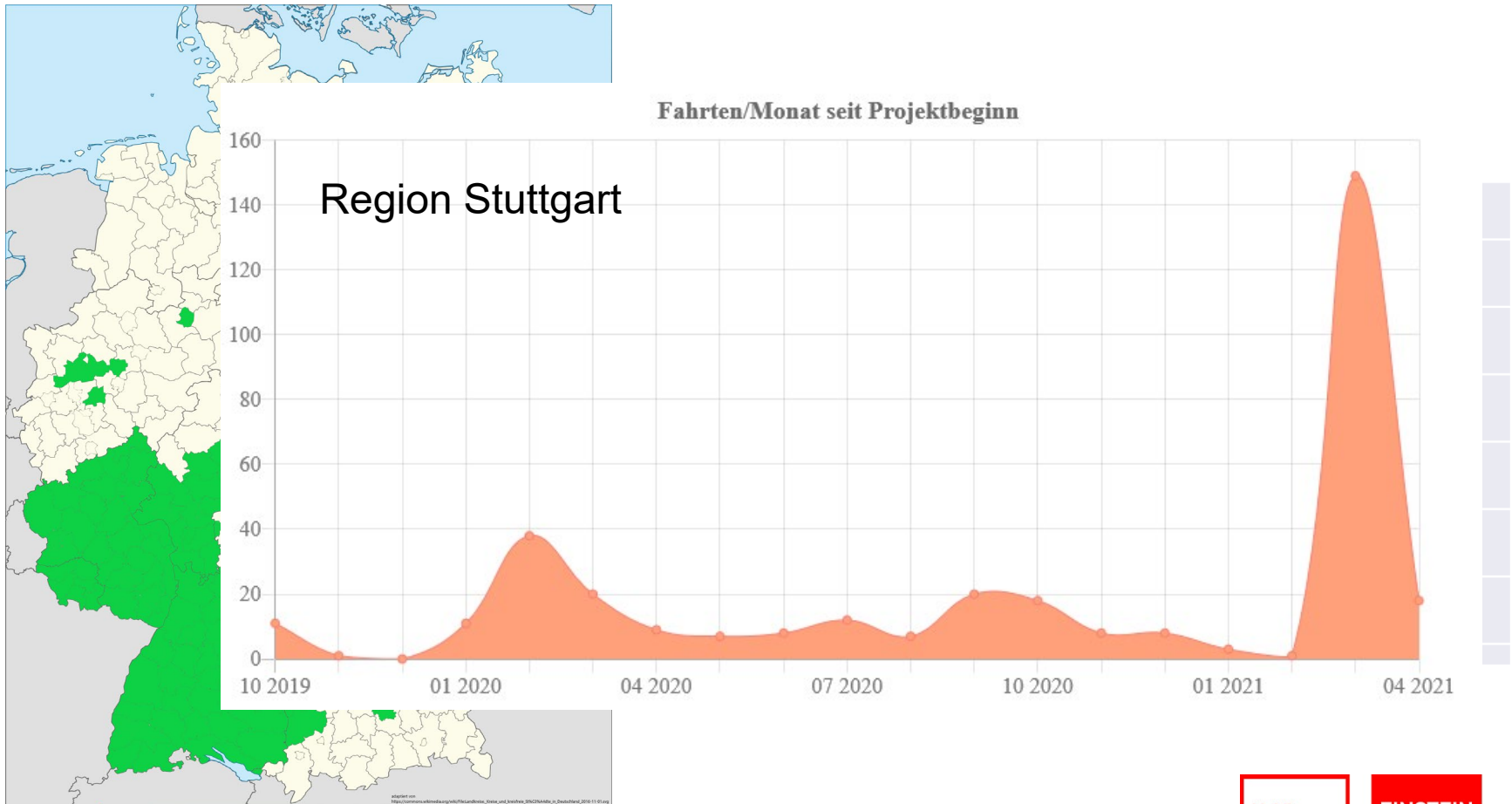
Abbrechen  Fertig 

© OpenStreetMap Mitwirkende Waldblick

AKTUELLER STAND

Aktueller Stand

App verfügbar für Android 6+ (seit März '19) und iOS 11+ (seit Mai '19)



ERGEBNISSE: BERLIN

<https://simra-project.github.io/>

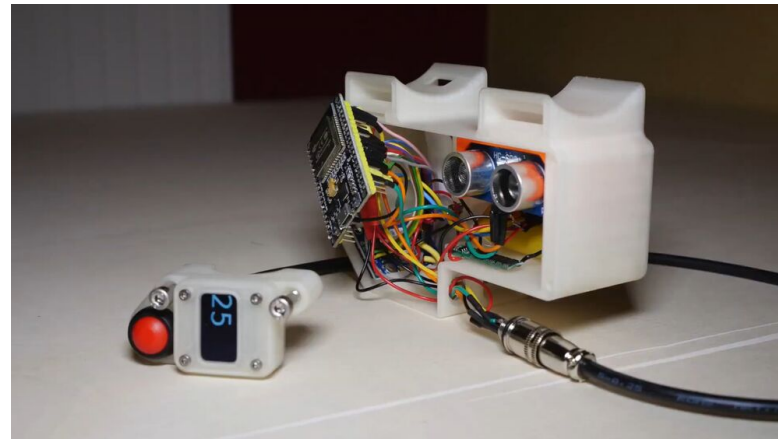
Ergebnisse

1. Es gibt nicht nur 5 Gefahrenschwerpunkte
2. Gute Fahrradinfrastruktur = wenig Probleme
3. Bestimmte Straßentypen fallen gehäuft auf:
 - Mehrspurig, ohne Radweg
 - Einspurig, ohne Radweg mit Parkstreifen: zu dichtes Überholen, Drängeln
 - Gewerbegebiet, Hochbordradweg: Abbiegesituationen bei Einfahrten
 - Nebenstraßen mit Abkürzungs- und Parksuchverkehr, Falschparkern
4. Zusatzgefahren durch Tramgleise

Erkenntnis

Erkennung von Beinaheunfällen funktioniert recht gut mit neuer ML-Erkennung... aber nur für die Fälle, wo es auch eine Bewegung gibt.

- ⇒ SimRa unterschätzt (massiv) zu dichtes Überholen
- ⇒ Teaser: OBS lässt sich per BLE mit SimRa-App koppeln



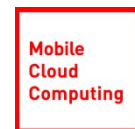
Thanks!

Questions?



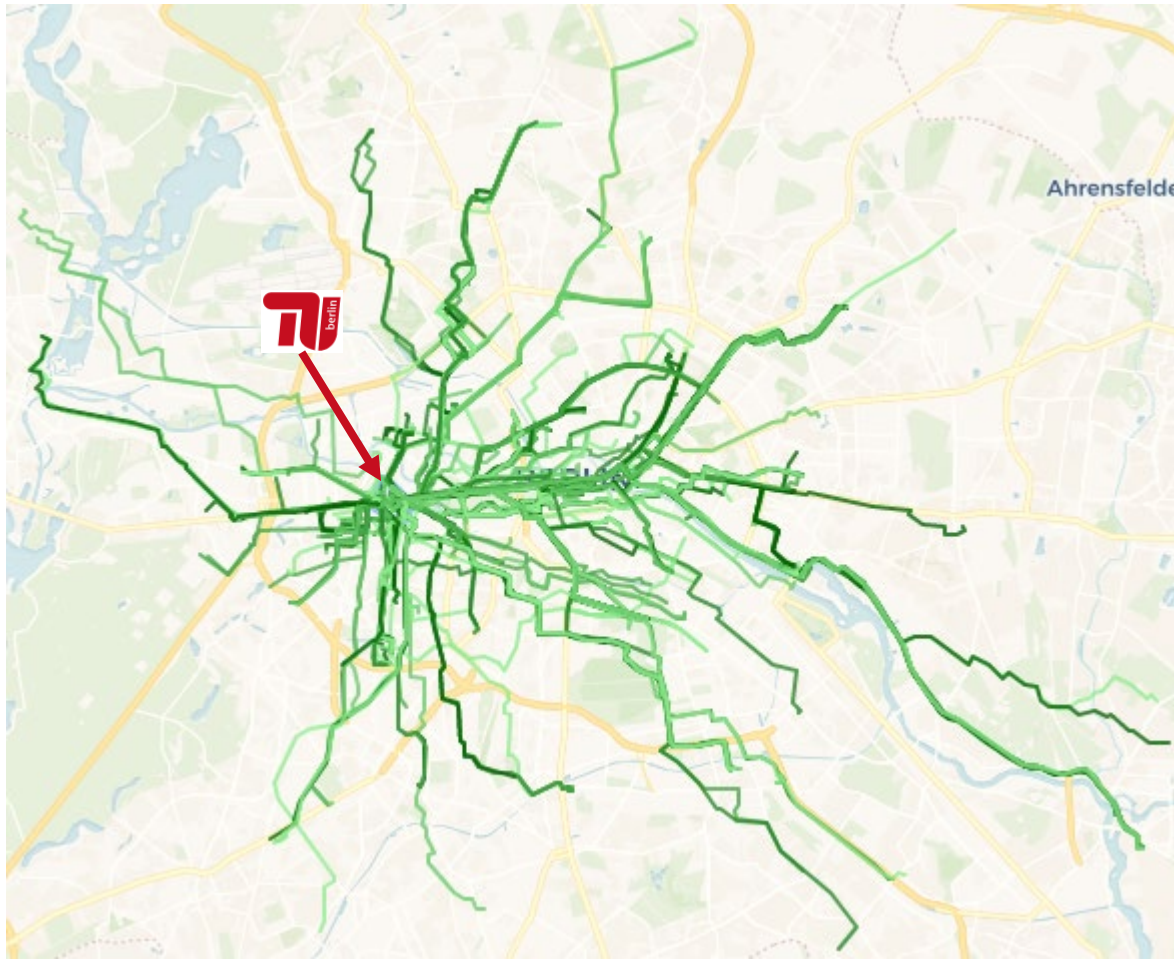
<https://simra-project.github.io/mitmachen.html>

Backup



First results

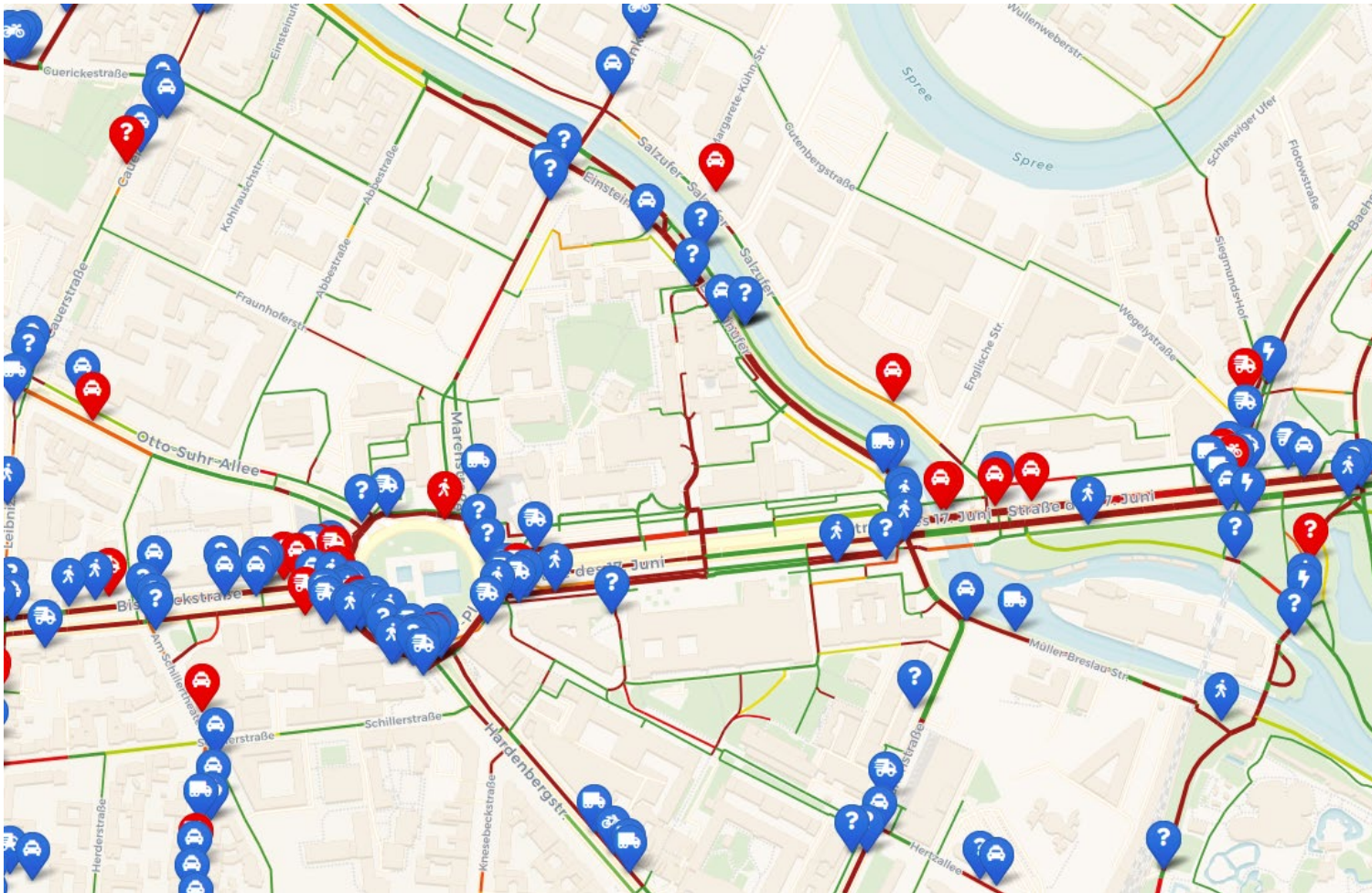
Rides to/from TU Berlin campus



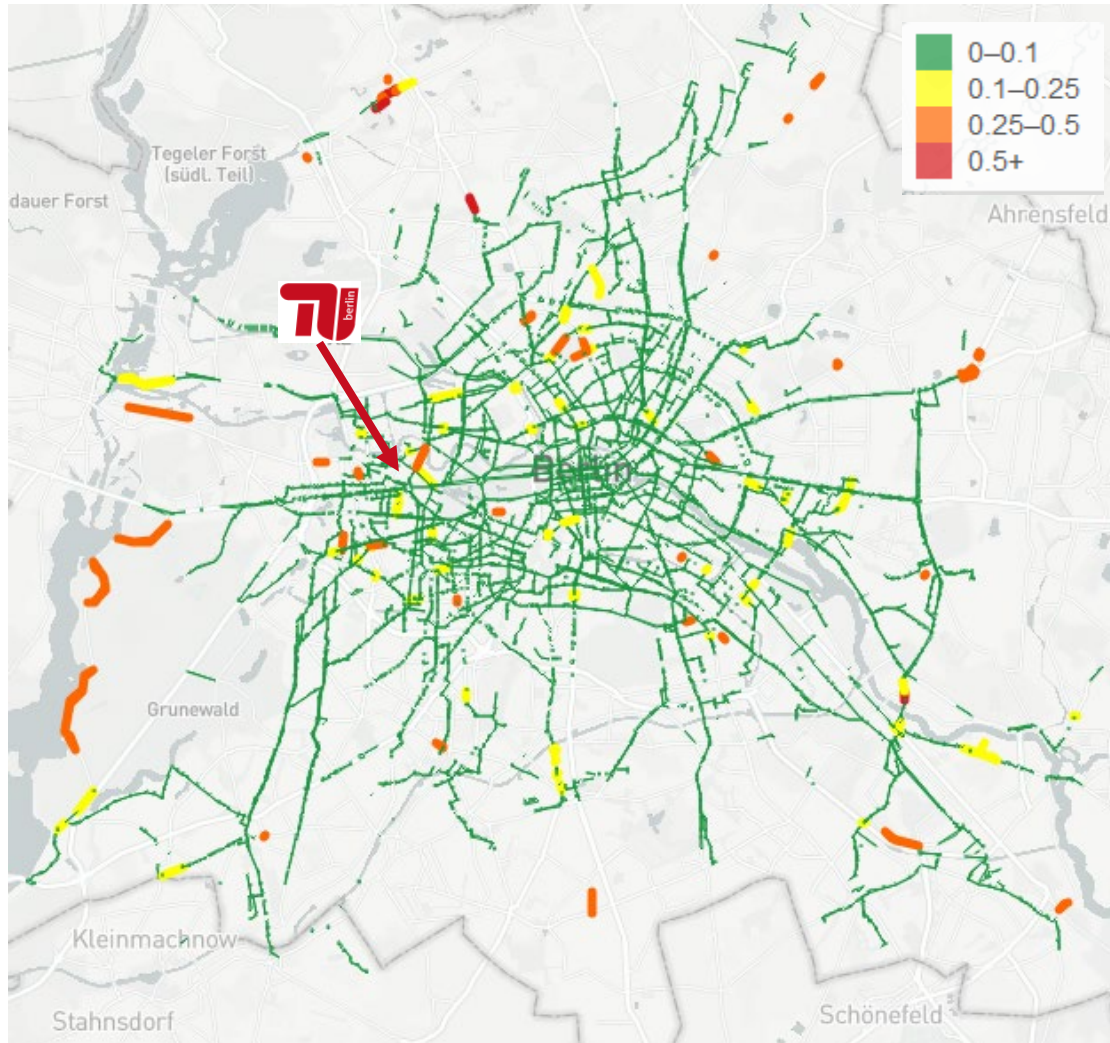
Rides to/from TU Berlin campus



Incidents around the TU Berlin campus



Dangerousness Berlin

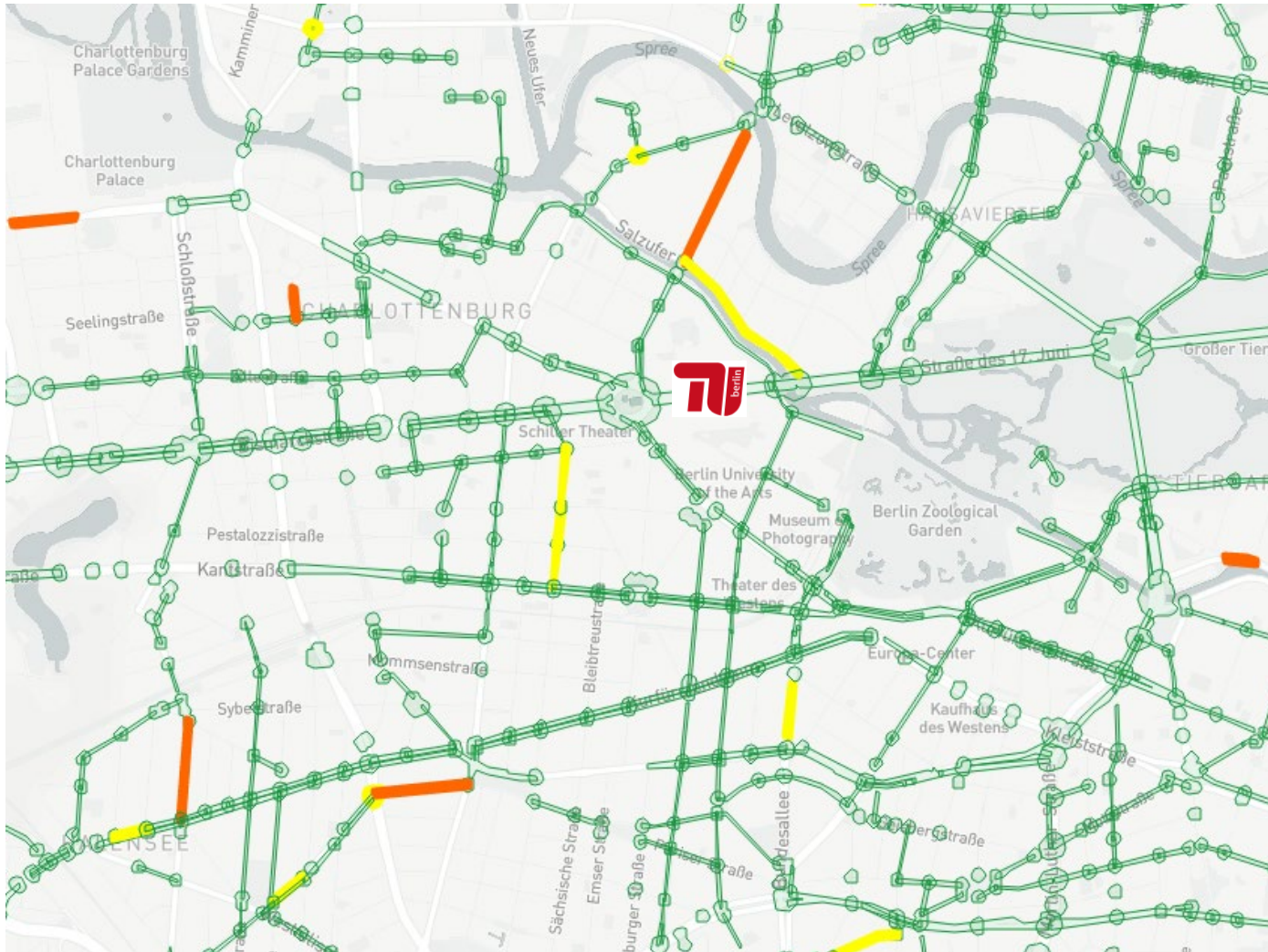


Score =
weighted #incidents/#rides

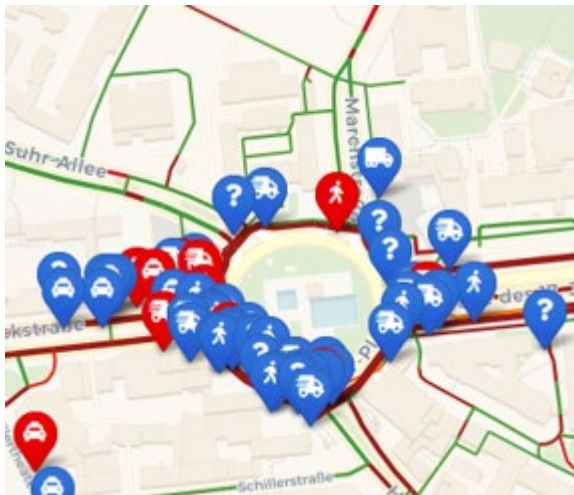
“scary” weight = 4.4

(thresholds of visualization colors
adapted for Berlin)

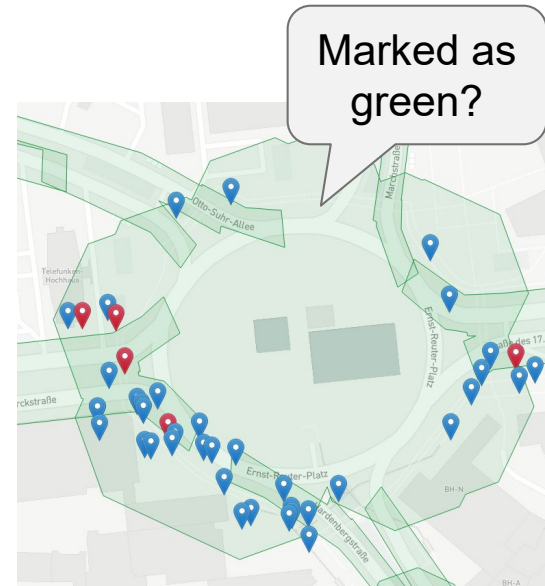
Dangerousness around the TU Berlin campus



Why is Ernst-Reuter-Platz relatively safe?

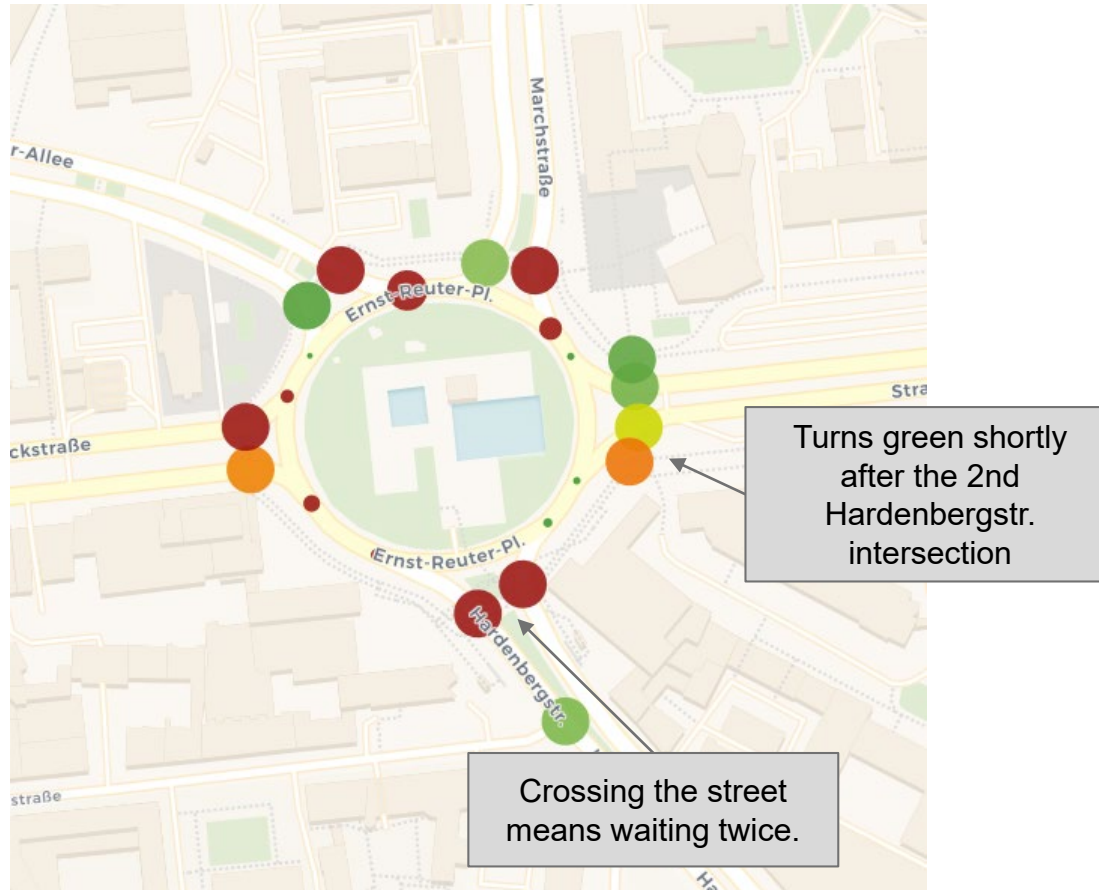


VS.



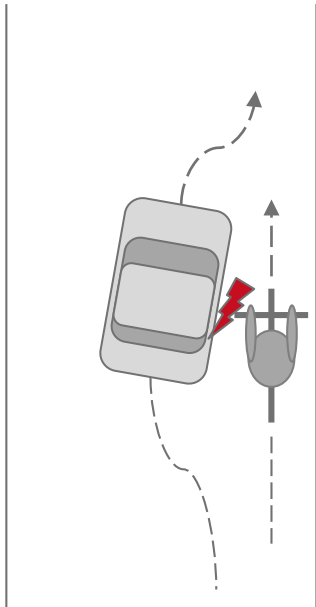
43 incidents – but also 1120 rides (data as of August 2020)
=> There is a problem in one out of 26 rides.

Wait time for intersections

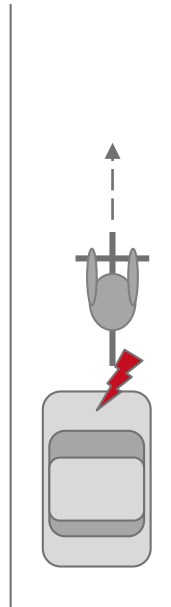


Detecting near miss incidents

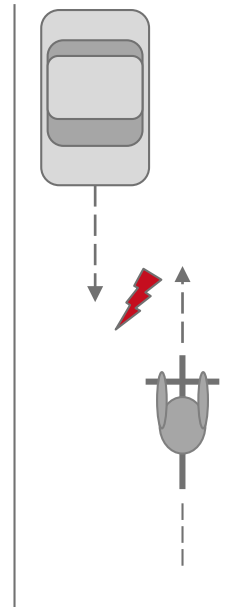
What is a near miss incident?



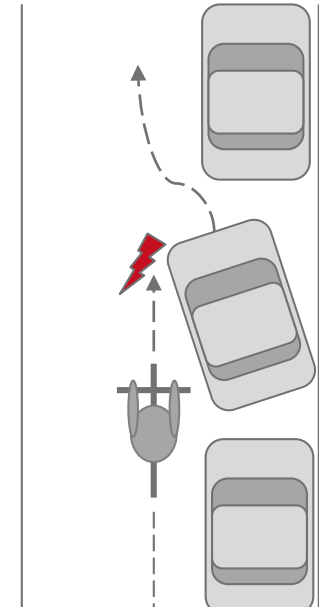
Close pass



Tailgating



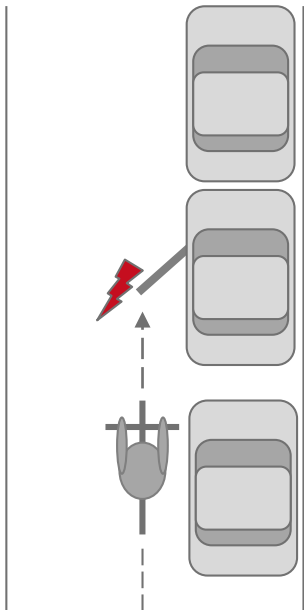
Approaching
head-on



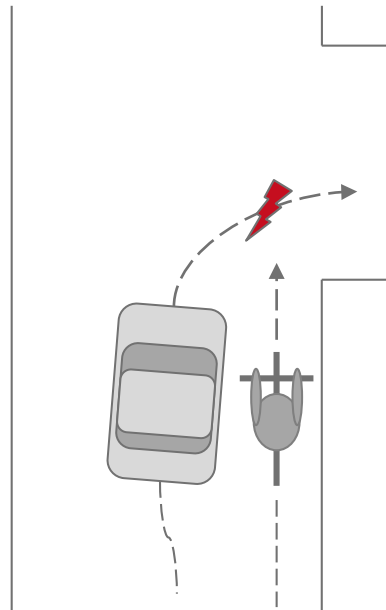
Pulling in or out

Based on: R. Aldred, S. Crossweller, Investigating the rates and impacts of near misses and related incidents among UK cyclists, Journal of Transport & Health, Elsevier 2015.

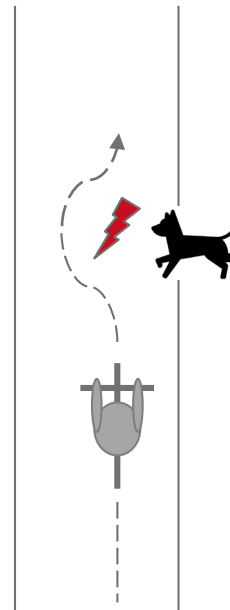
What is a near miss incident? (cont.)



Near-dooring



Left/right hook



Dodging an obstacle

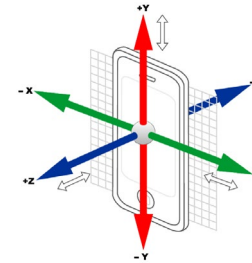


Other

Based on: R. Aldred, S. Crossweller, Investigating the rates and impacts of near misses and related incidents among UK cyclists, Journal of Transport & Health, Elsevier 2015.

Core Idea and Approach

Sudden peaks in accelerometer data are a good proxy for near miss incidents



GPS traces

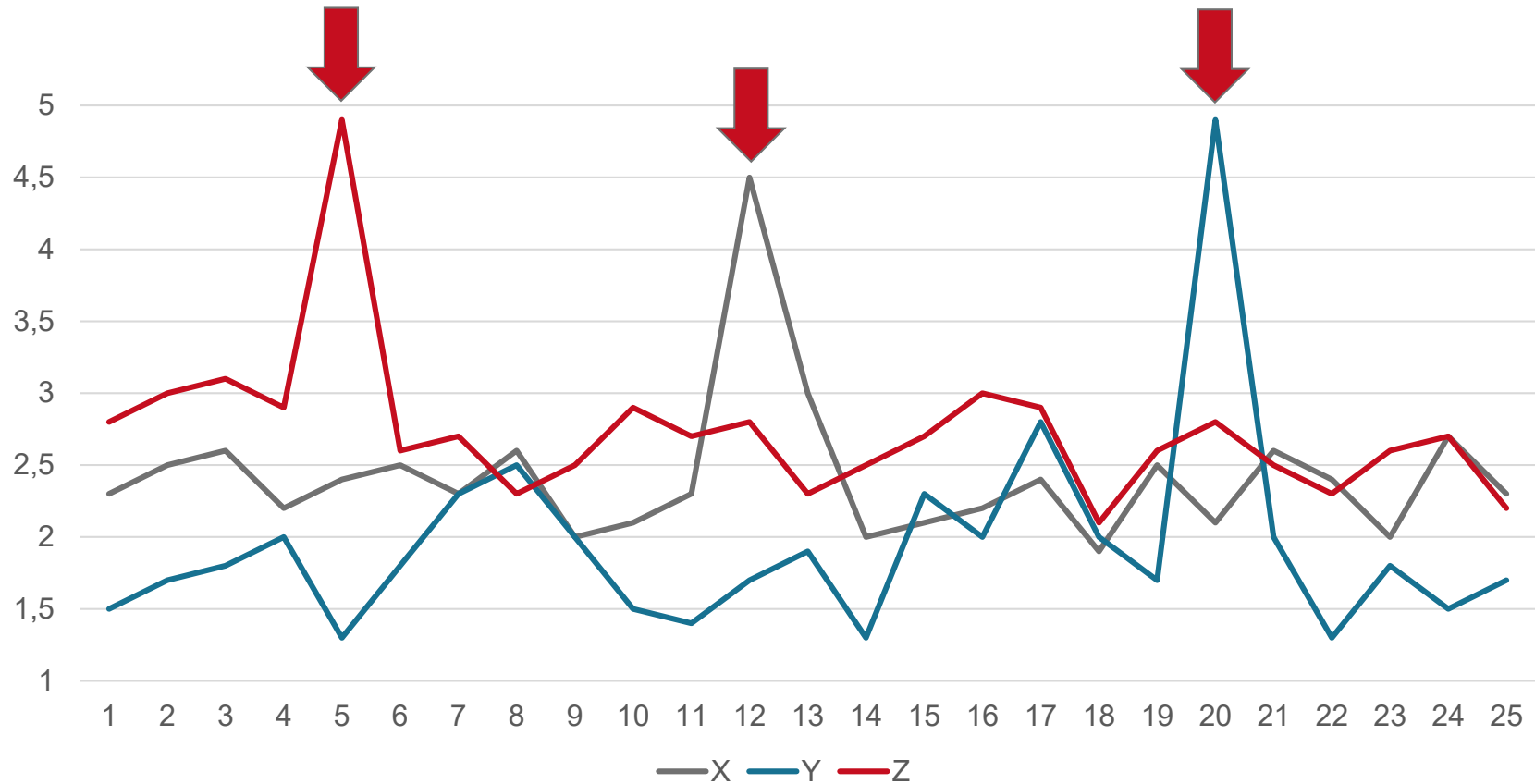
- to identify main routes
- to calculate relative “dangerousness” score

Measurements are imperfect so that we ask users to annotate their ride.



A first heuristic

“pick the biggest peaks“



Problems with the first heuristic

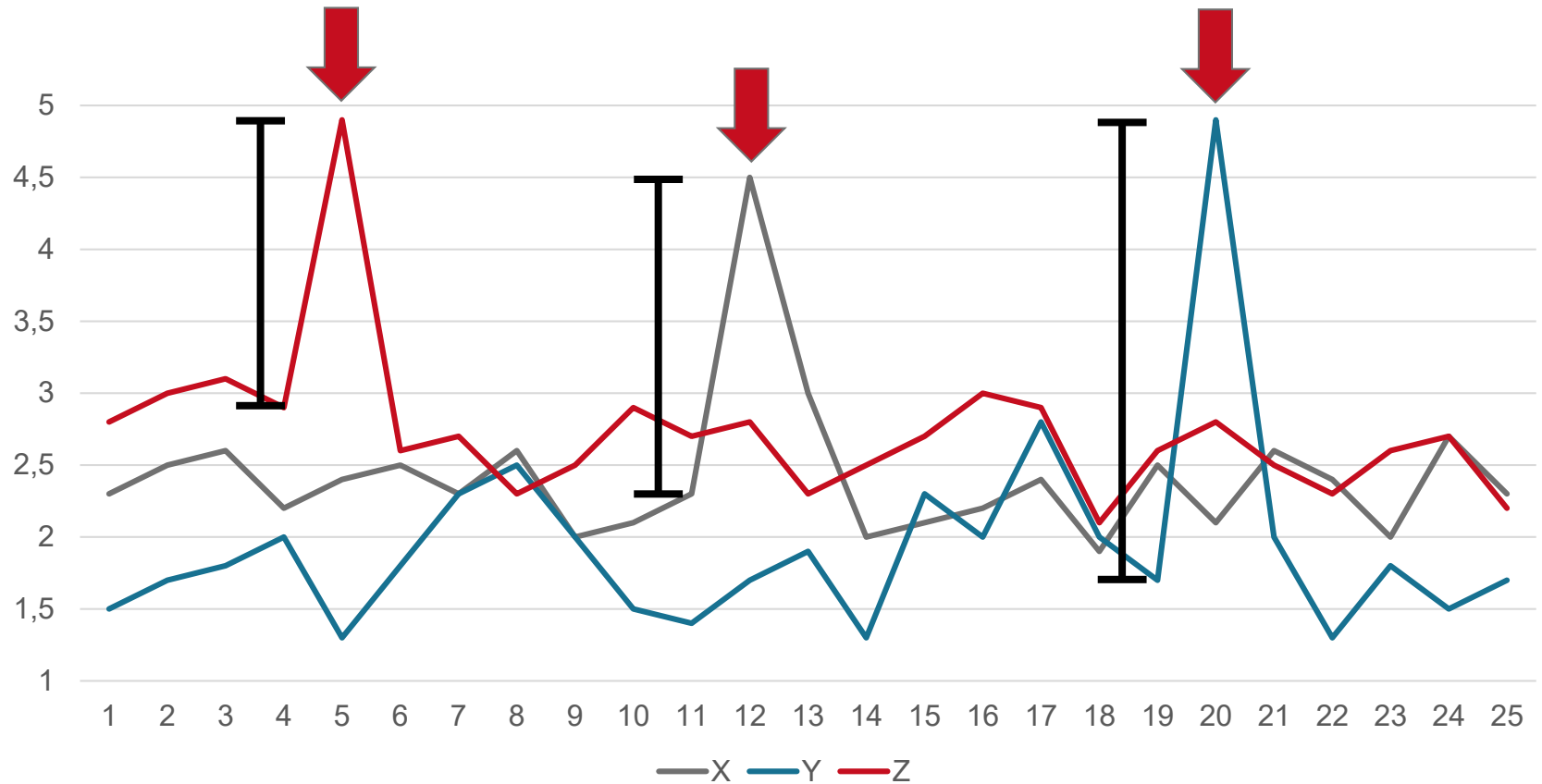
Cobblestone roads max out the sensor.

=> Use biggest jump instead:

1. Create 1s buckets
2. Find max and min per bucket
3. Calculate diff = max – min
4. Over all buckets pick $\max\{\text{diff}\}$

A second heuristic

“pick the two biggest jumps per dimension”



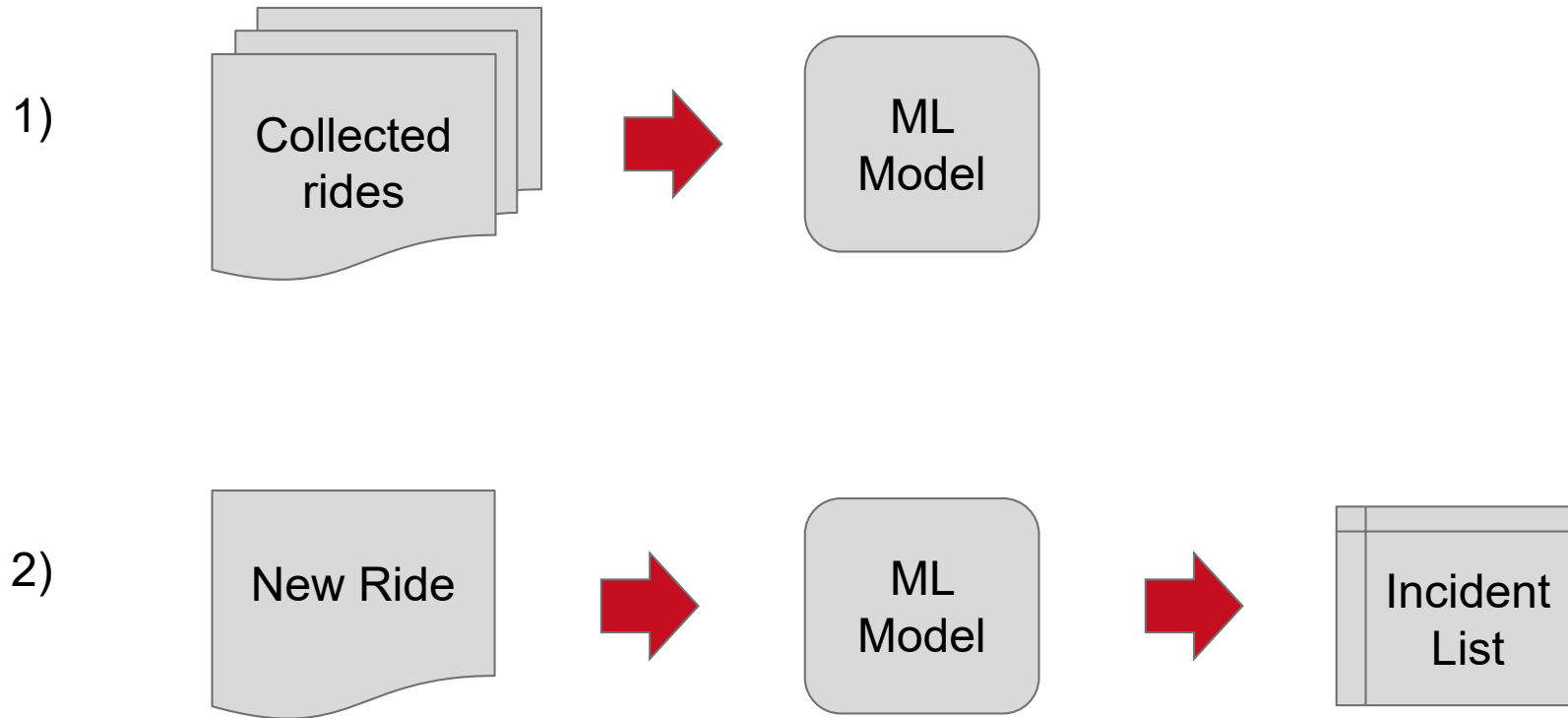
Problems with the second heuristic

Has been live for ca. one year

- Works well for grandma-style cycling
- Detects traffic lights and potholes otherwise

=> try machine learning

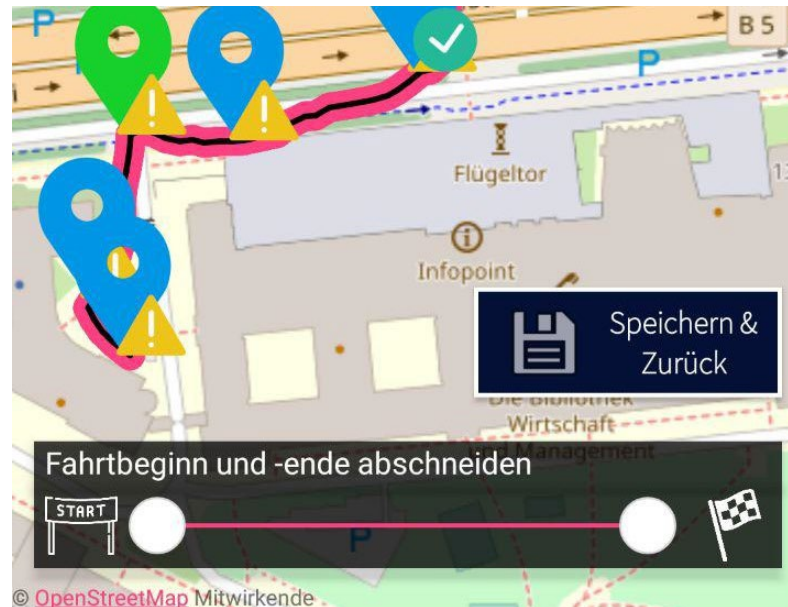
ML-based approach



Preserving user privacy

Privacy slider

- Only the cropped ride is stored locally after editing
- Upload needs to be triggered manually *after* the editing



Per-record pseudonymization

