

# Indoor-Localisation unter Android

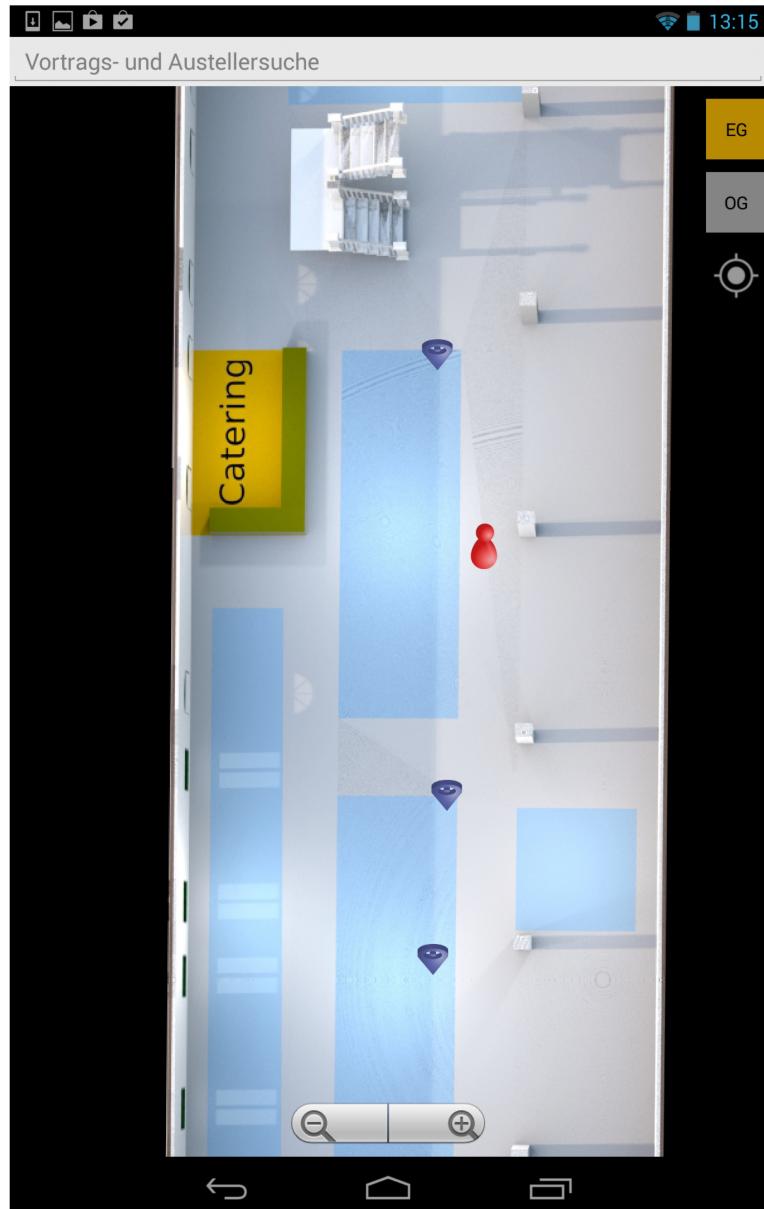


Robert Linden <r.linden@tarent.de>

→ tarent

# So sieht sie aus:

→ tarent



## MOTIVATION

- Bestimmung der Besucherposition
- Nutzung bestehender WLAN-Netze
- Nutzung möglichst vieler Sensoren (Sensor Fusion)
- Hohe Genauigkeit (< 3m)
- Suche & Anzeige von Points of Interest
- Automatische Etagenerkennung  
... und das alles mit Open Source Software!

<https://github.com/tarent/invio>



## Wie machen wir Indoor Navigation?

Welche Komponenten verwenden wir?

Sensoren

WLAN

Bewegungssensoren

Fusion aller Sensoren

Matching auf Umgebungsinformationen

Ausblick & Demo

→ **initialer Download von Umgebungsdaten**

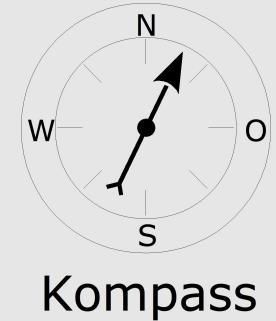
→ Lokale Sensorabfrage

→ Lokale Positionsberechnung

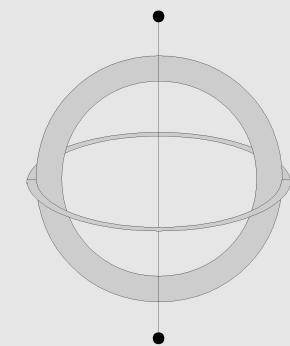
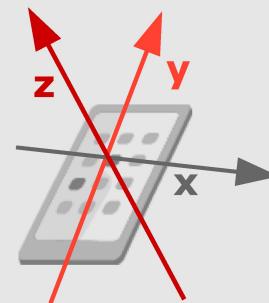
Karte inkl. WLAN-Vermessung



- initialer Download von Umgebungsdaten



## → Lokale Sensorabfrage



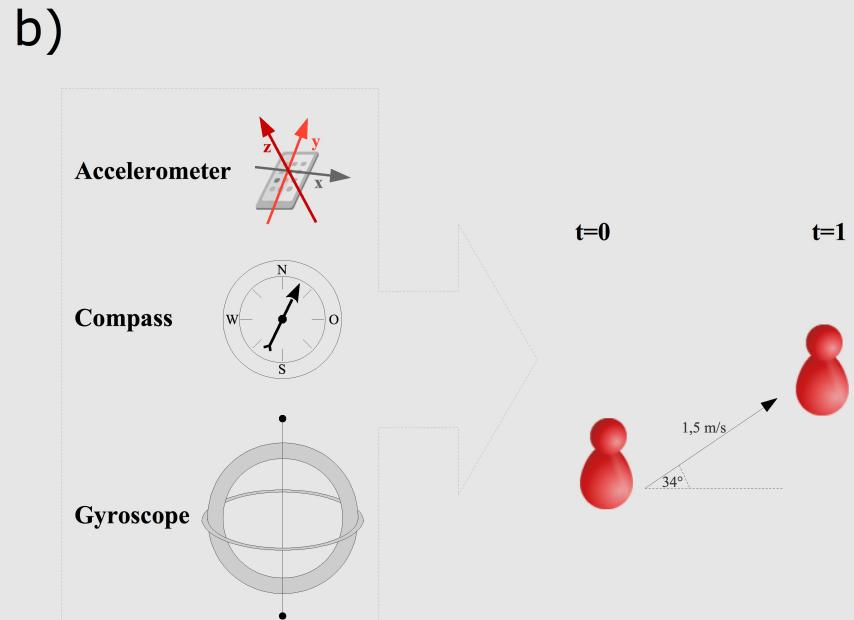
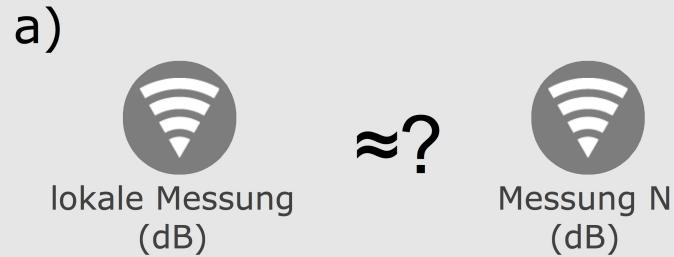
- Lokale Positionsberechnung

# Wie?

→ initialer Download  
von Umgebungsdaten

→ Lokale Sensorabfrage

→ Lokale Positionsberechnung



# Agenda

Wie machen wir Indoor Navigation?



Welche Komponenten verwenden wir?

Sensoren

WLAN

Bewegungssensoren

Fusion aller Sensoren

Matching auf Umgebungsinformationen

Ausblick & Demo

# Komponenten

→ tarent

→ mapeditor

→ mapserver

→ admin-app

→ customer-app

# Komponenten

→ tarent

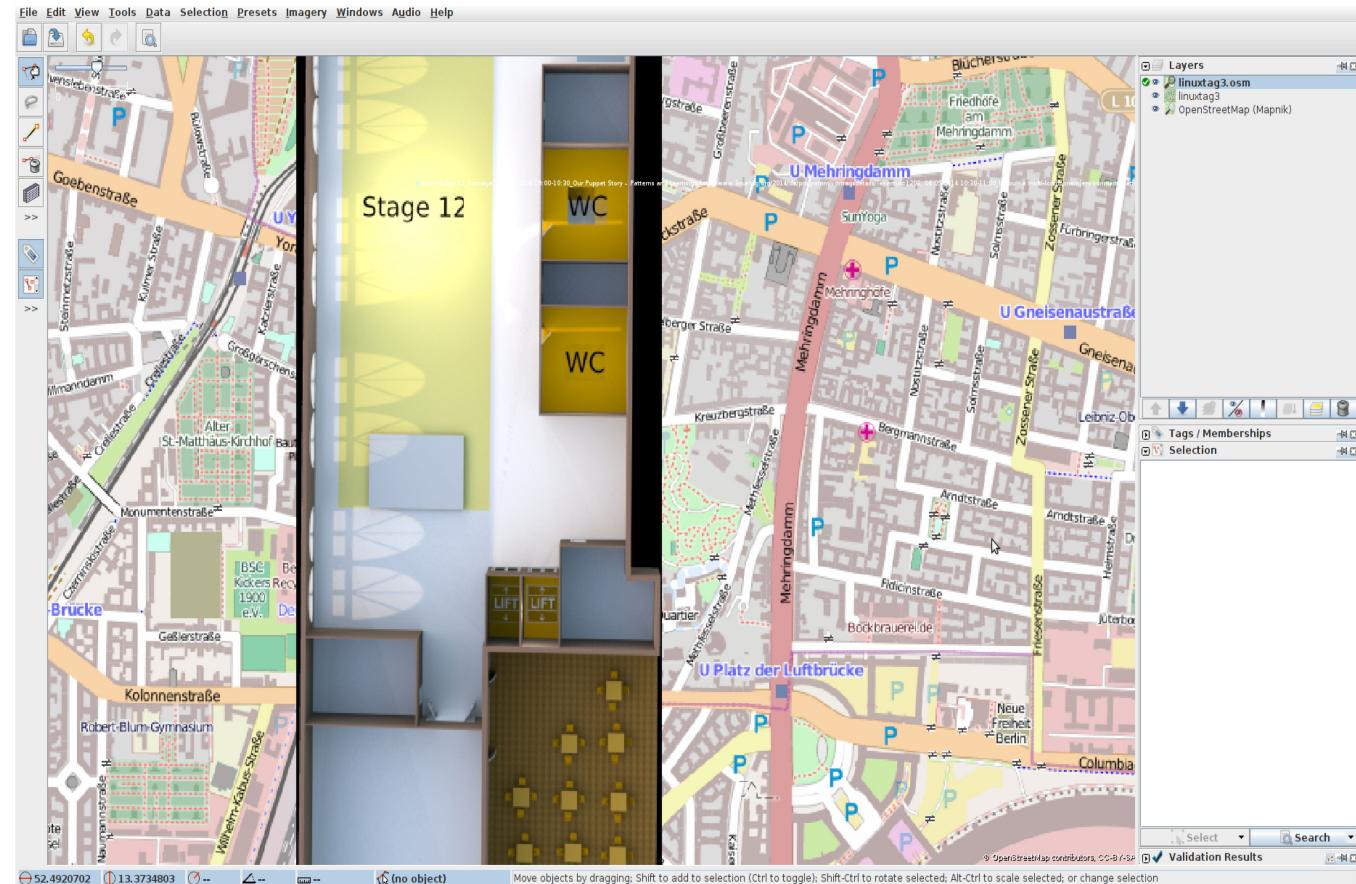
→ mapeditor

- Erstellen von georeferenzierten Karten inkl. Metadaten
- Hinterlegen von georeferenzierten Objekten
- Upload zum mapserver

→ mapserver

→ admin-app

→ customer-app



# Komponenten

→ mapeditor

→ mapserver

→ Erstellen von Karten-Tiles unterschiedlicher Zoomstufen

→ Bereitstellung von Karten inkl. aller Metadaten aus dem mapeditor

→ Bereitstellung der initialen WLAN-Vermessung aus der **admin-app**

→ admin-app

# Komponenten

→ tarent

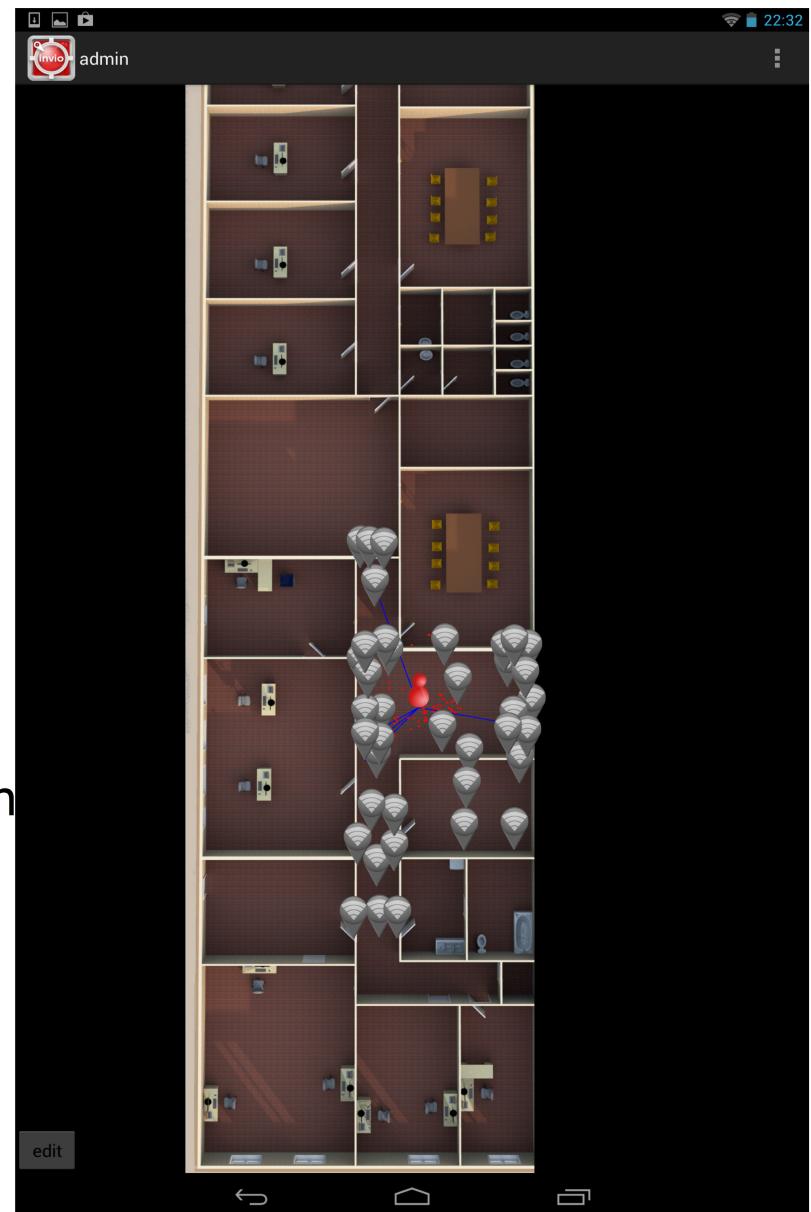
→ mapeditor

→ mapserver

→ admin-app

→ Erstellen von initialen WLAN-Messungen mit Georeferenzierung je Karte

→ customer-app



# Komponenten

→ tarent

→ mapeditor

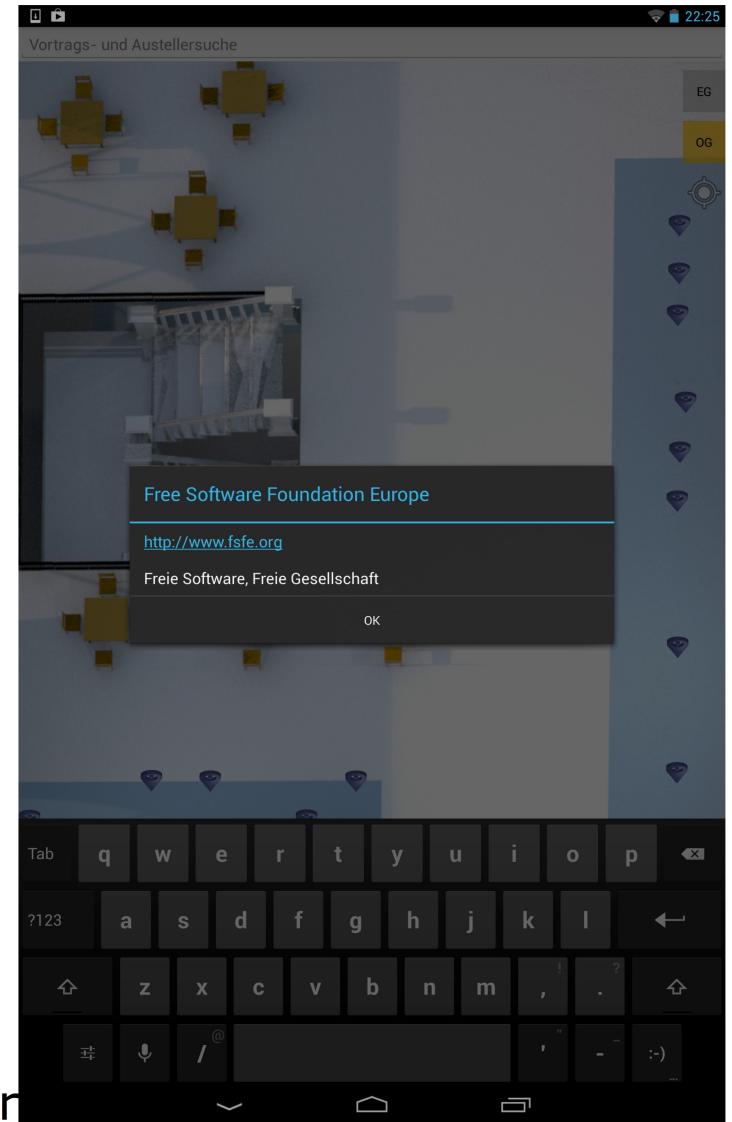
→ mapserver

→ admin-app

→ customer-app

→ initiale Abfrage der Karten vom mapserver

→ Lokalisierung durch Matching von aktuellen Sensormessdaten auf die Karten-Daten



# Agenda

Wie machen wir Indoor Navigation?

Welche Komponenten verwenden wir?

Sensoren



WLAN

Bewegungssensoren

Fusion aller Sensoren

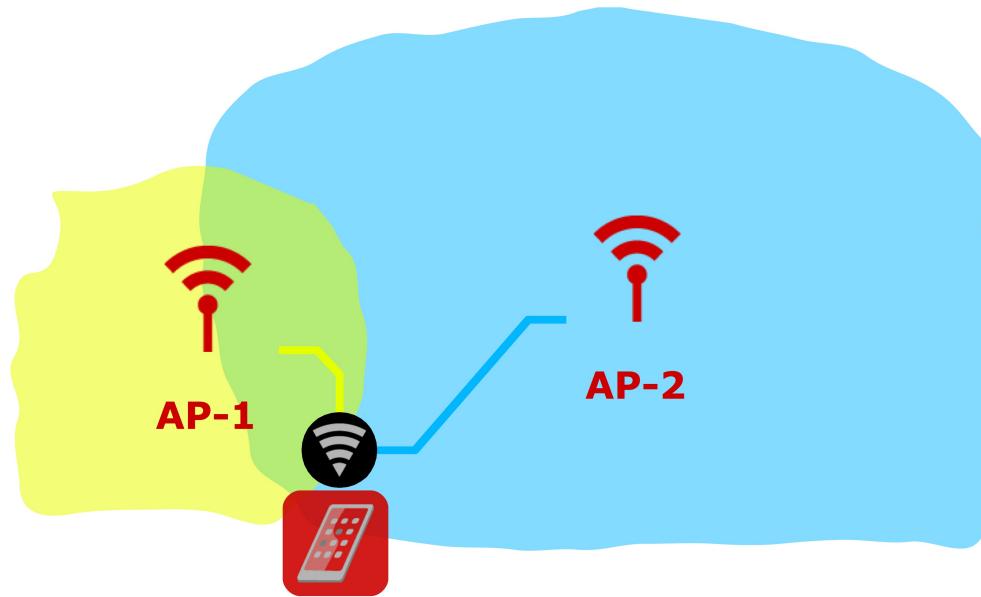
Matching auf Umgebungsinformationen

Ausblick & Demo

# Fingerprint

- Ergebnis eines WLAN Broadcast Callbacks sind die MAC-Adressen aller sichtbaren Access-Points und deren empfangene Signalstärke:

```
"id": "FP-now"  
"AP-1": { "-45" }  
"AP-2": { "-87" }
```

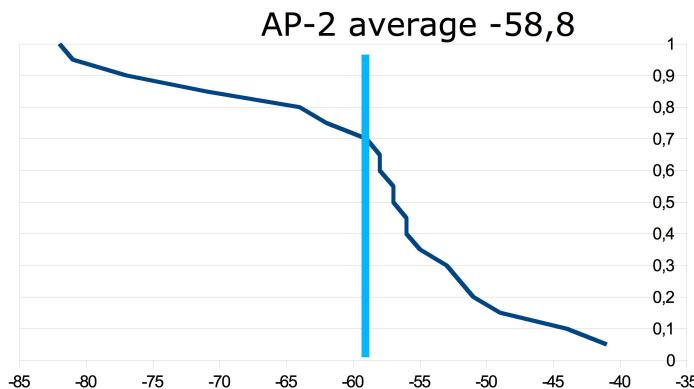
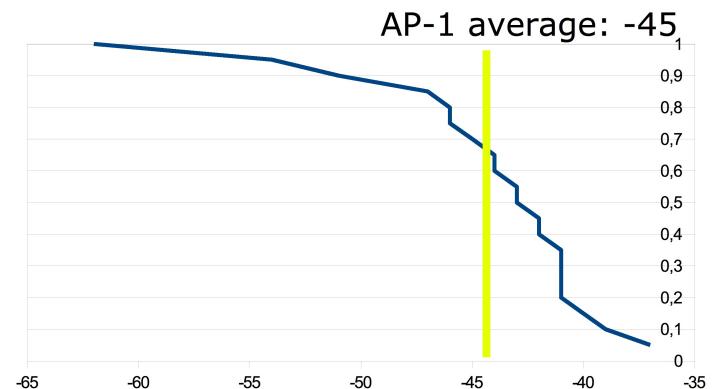


# Histogramme

- Innerhalb eines WLAN-Scavorgange werden 5 Fingerprint Messungen durchgeführt, die in einem gewichteten Histogramm abgelegt werden:

"histogram":

```
"point": {"mAltitude": 0, "mLatitudeE6": 50722458, "mLongitudeE6": 7061819}
"AP-1": { "-45": 1.0 }
"AP-2": { "-65": 0.2, "-60": 0.2, "-58": 0.4, "-53": 0.2 }
```

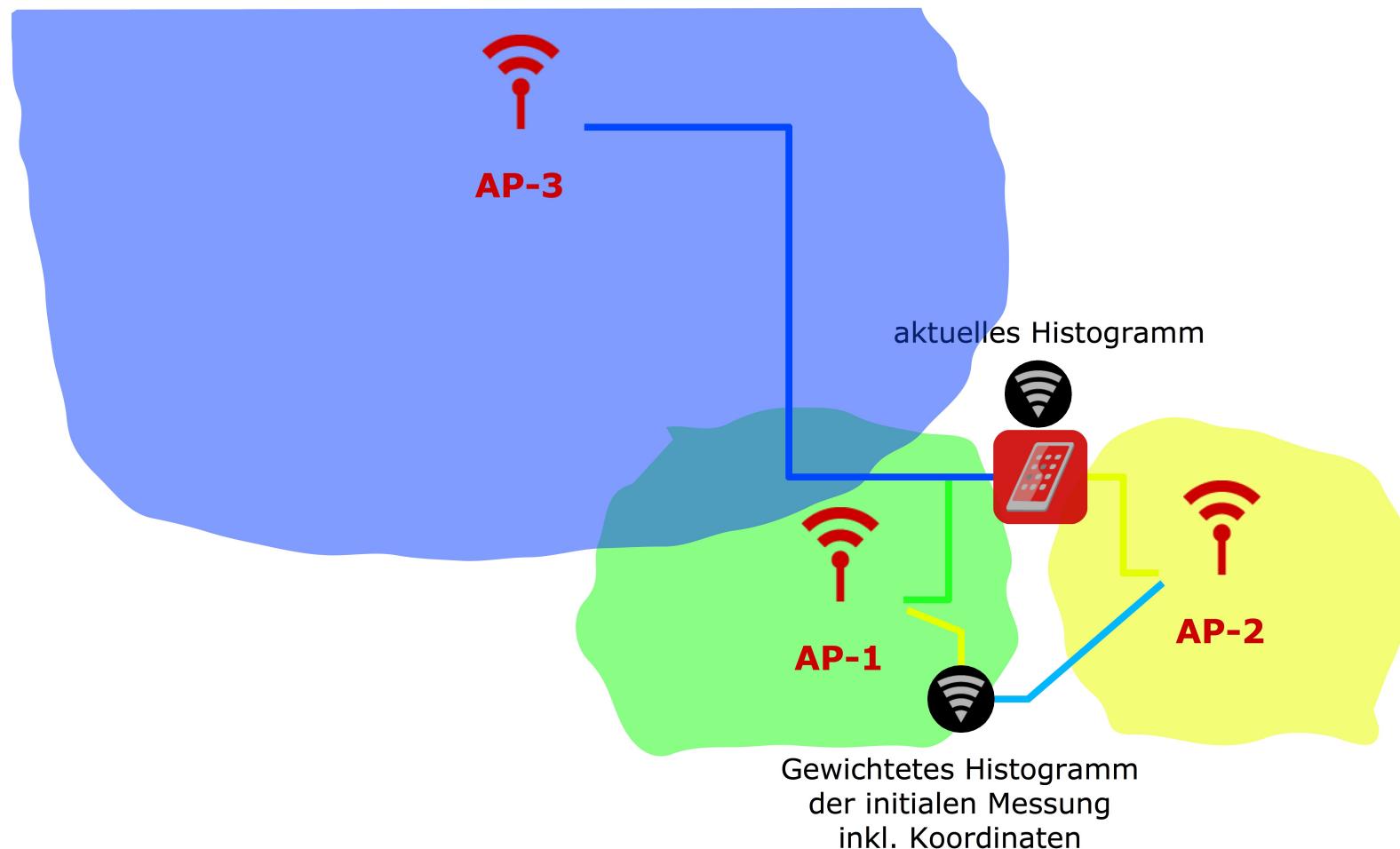


- Histogramme werden verwendet um Schwankungen ausgleichen zu können.
- Der Konfigurations-Spielraum ist gering, da langsame Geräte bis zu 5 Sekunden pro Messung benötigen, schnelle nur 1 Sekunde.

# Signals Map Overlay

→ tarent

- Die **customer-app** vermisst in kurzen Abständen die aktuelle WLAN-Umgebung und speichert sie in einem temporären Histogramm.

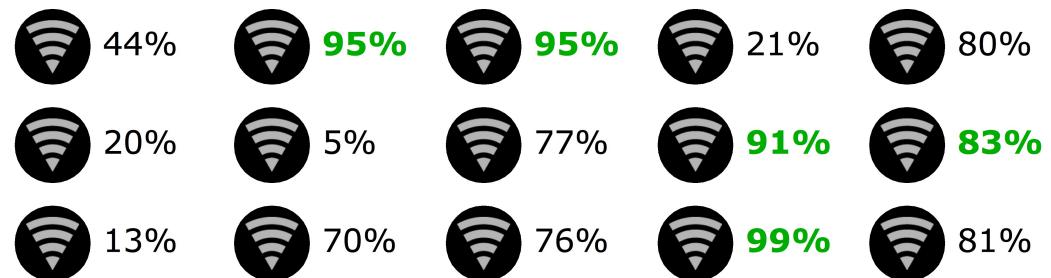


- Vergleich des aktuellen Histogramms mit der Histogramm-Datenbank
  - Kullback-Leibler Divergenz
  - ?
  
- Es werden die N (z.B. N=5) *ähnlichsten* Histogramme ermittelt.

**aktuelles Histogramm**

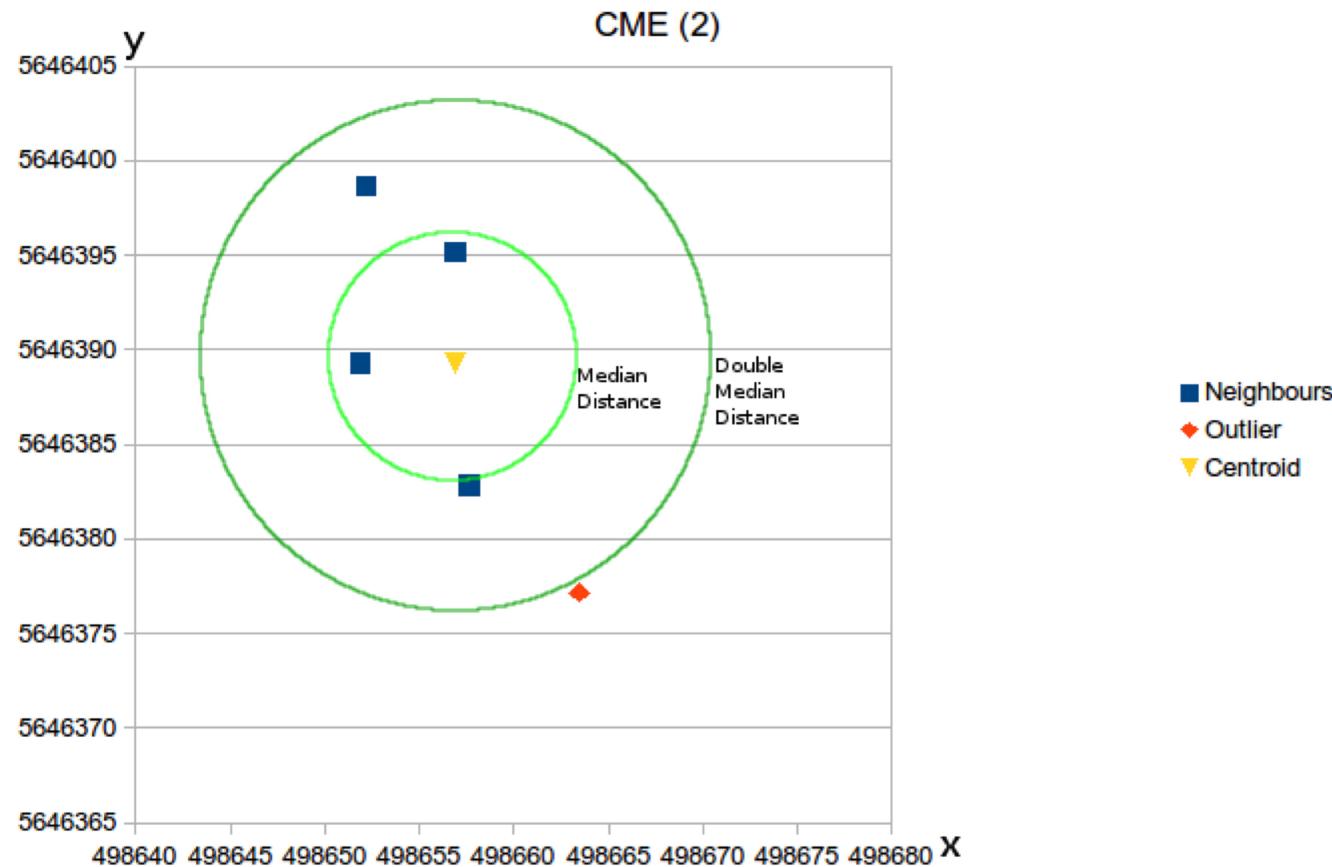


**Histogramm Datenbank**



# Outlier Elimination

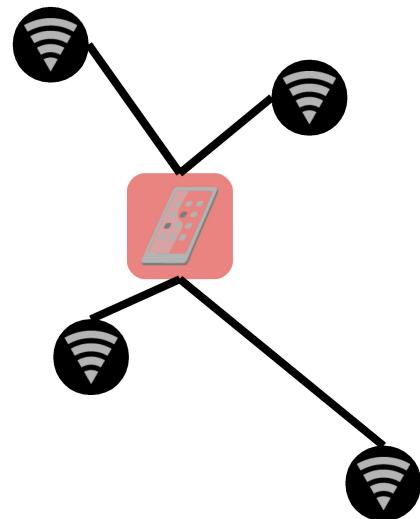
- Basierend auf dem Centroid, der aus den Positionen der stärksten 5 Histogramme berechnet wird



- Der Faktor 2 für den Schwellwert hat sich als ausgewogen erwiesen

# Interpolierung der Position

- Interpolation der Position aus den Histogrammen, die nach der Outlier Elimination übrig geblieben sind.



- In einem Sliding Window kann z.B. von den 5 letzten Positionen der Median verwendet werden um die Genauigkeit und Outlier Elimination weiter zu verbessern.

# Agenda

Wie machen wir Indoor Navigation?

Welche Komponenten verwenden wir?

Sensoren

WLAN

Bewegungssensoren

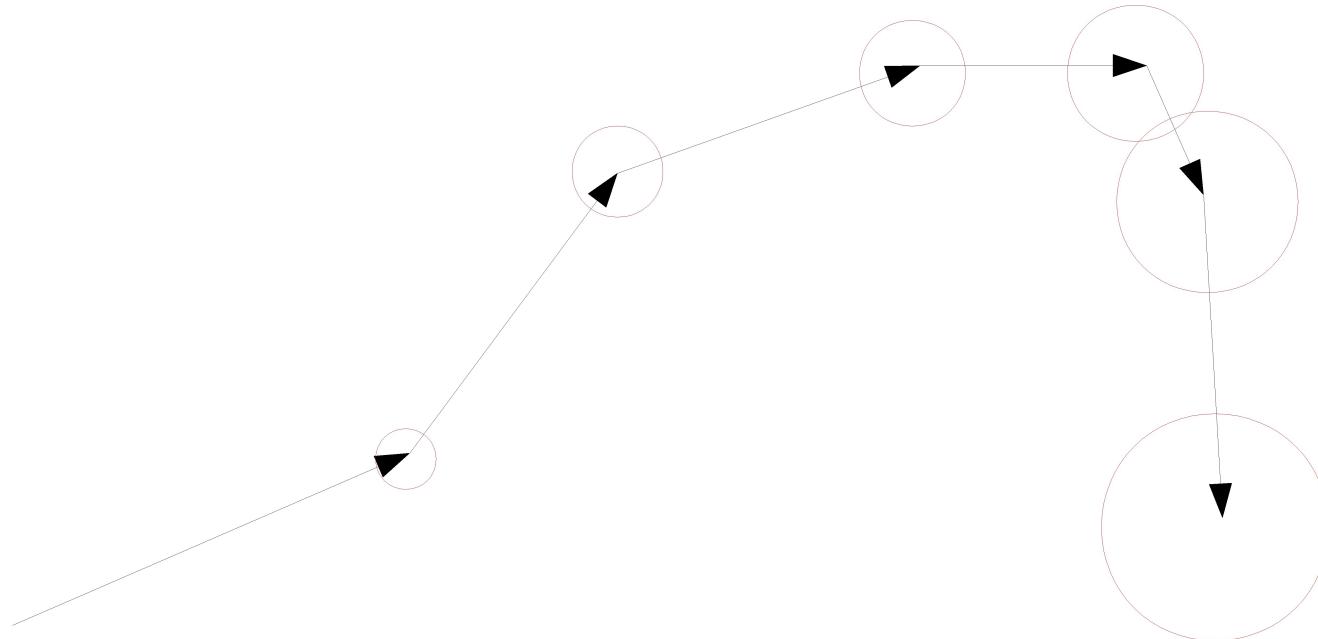
Fusion aller Sensoren

Matching auf Umgebungsinformationen

Ausblick & Demo

## Fortlaufende Positionsabschätzung mittels:

- Bewegungsrichtung
- Geschwindigkeit



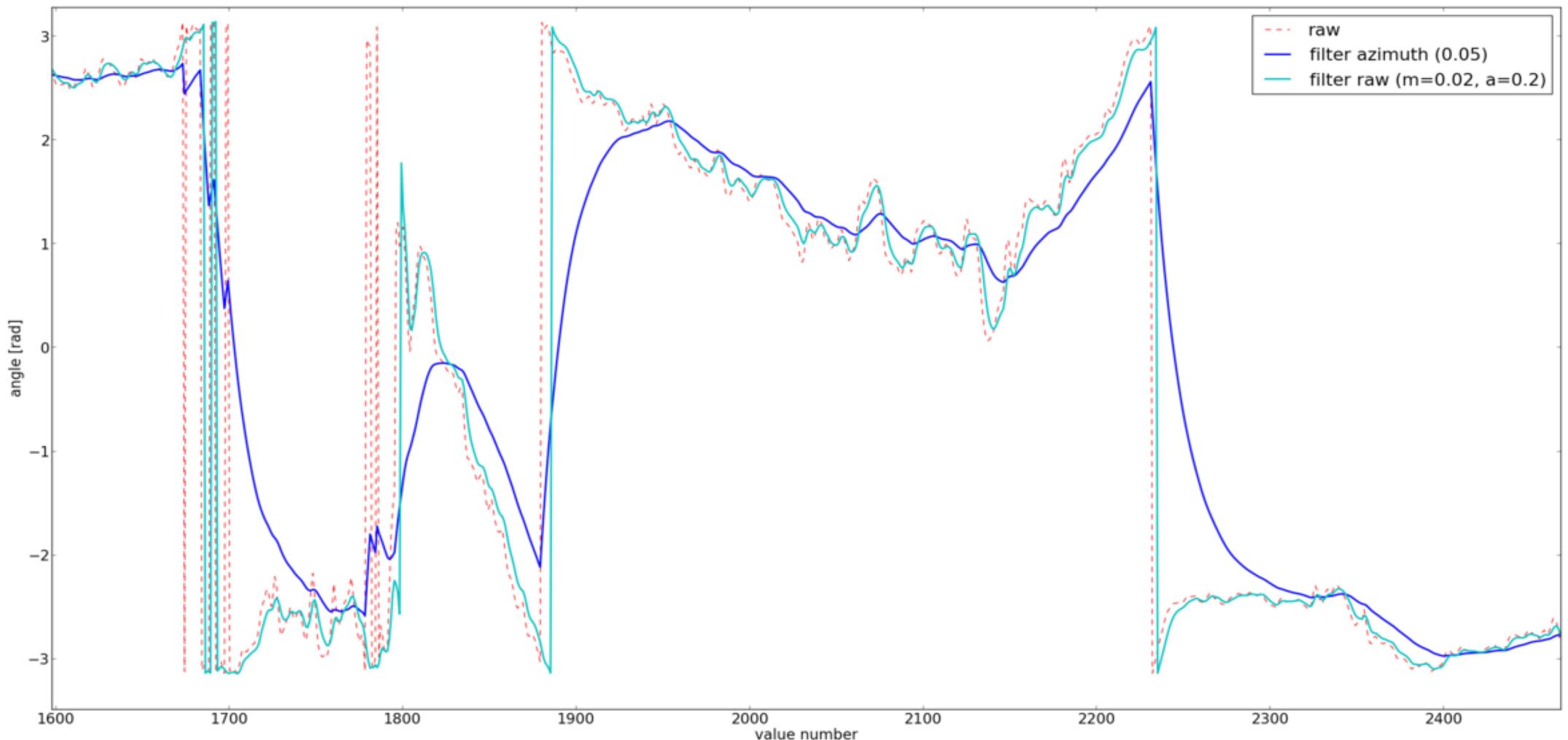
## Bewegungsrichtung

- Kompass
- Sensor.TYPE\_MAGNETIC\_FIELD
- Sensor.TYPE\_ACCELEROMETER

## Geschwindigkeit

- Sensor.TYPE\_LINEAR\_ACCELERATION (wenn verfügbar)
- Im einfachsten Fall binär:
  - Stehen
  - Gehen, mit konstanter Geschwindigkeit
- Besser: Schrittfrequenz zählen

## Tiefpassfilter hilft gegen starkes Sensorrauschen



# Agenda

Wie machen wir Indoor Navigation?

Welche Komponenten verwenden wir?

Sensoren

WLAN

Bewegungssensoren

Fusion aller Sensoren

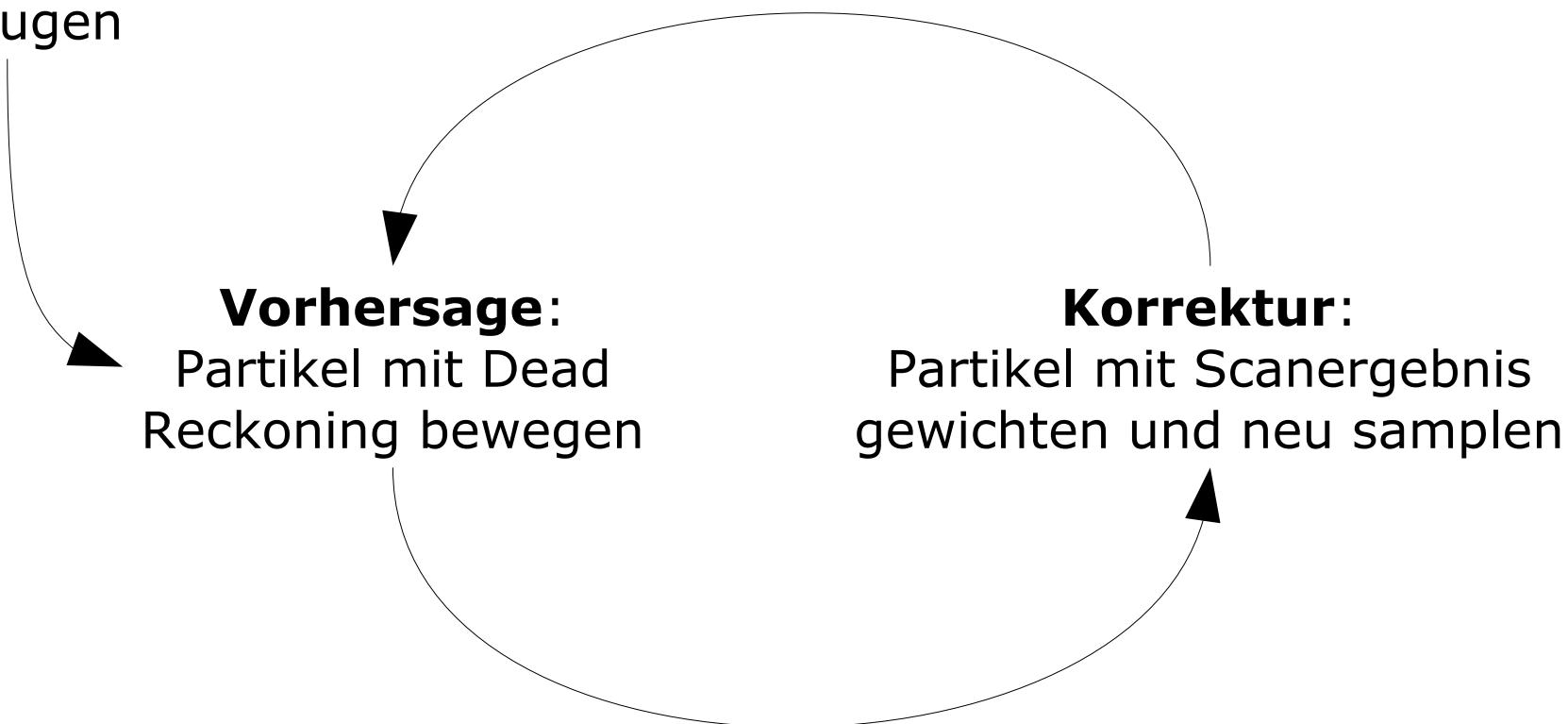
Matching auf Umgebungsinformationen

Ausblick & Demo

Informationen aus WLAN-Messung und Bewegungsabschätzung werden durch Partikelfilter kombiniert:

## **Initialisierung:**

Partikel  
erzeugen



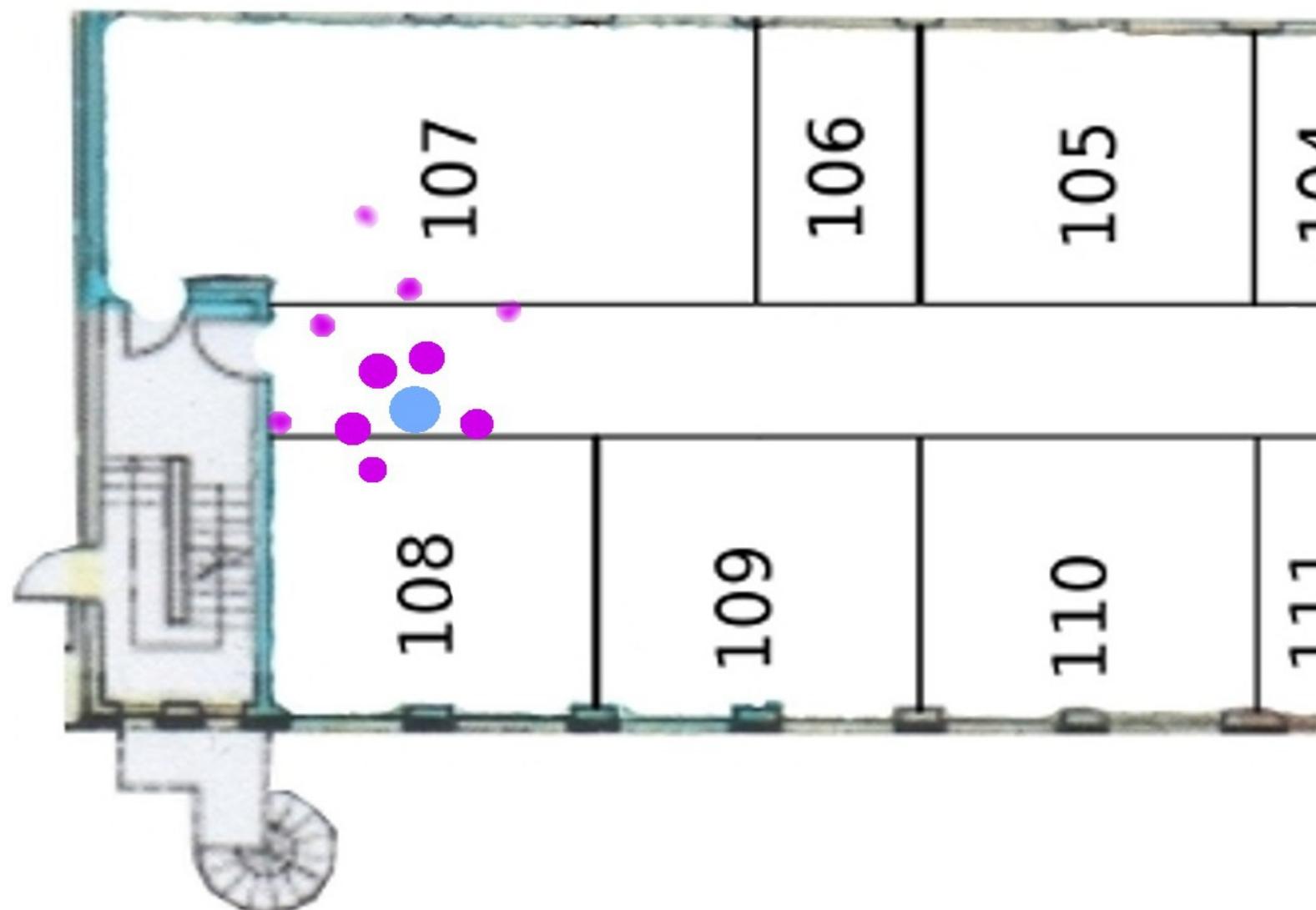
Partikel werden um den angenommenen Startpunkt herum durch Streuung erzeugt



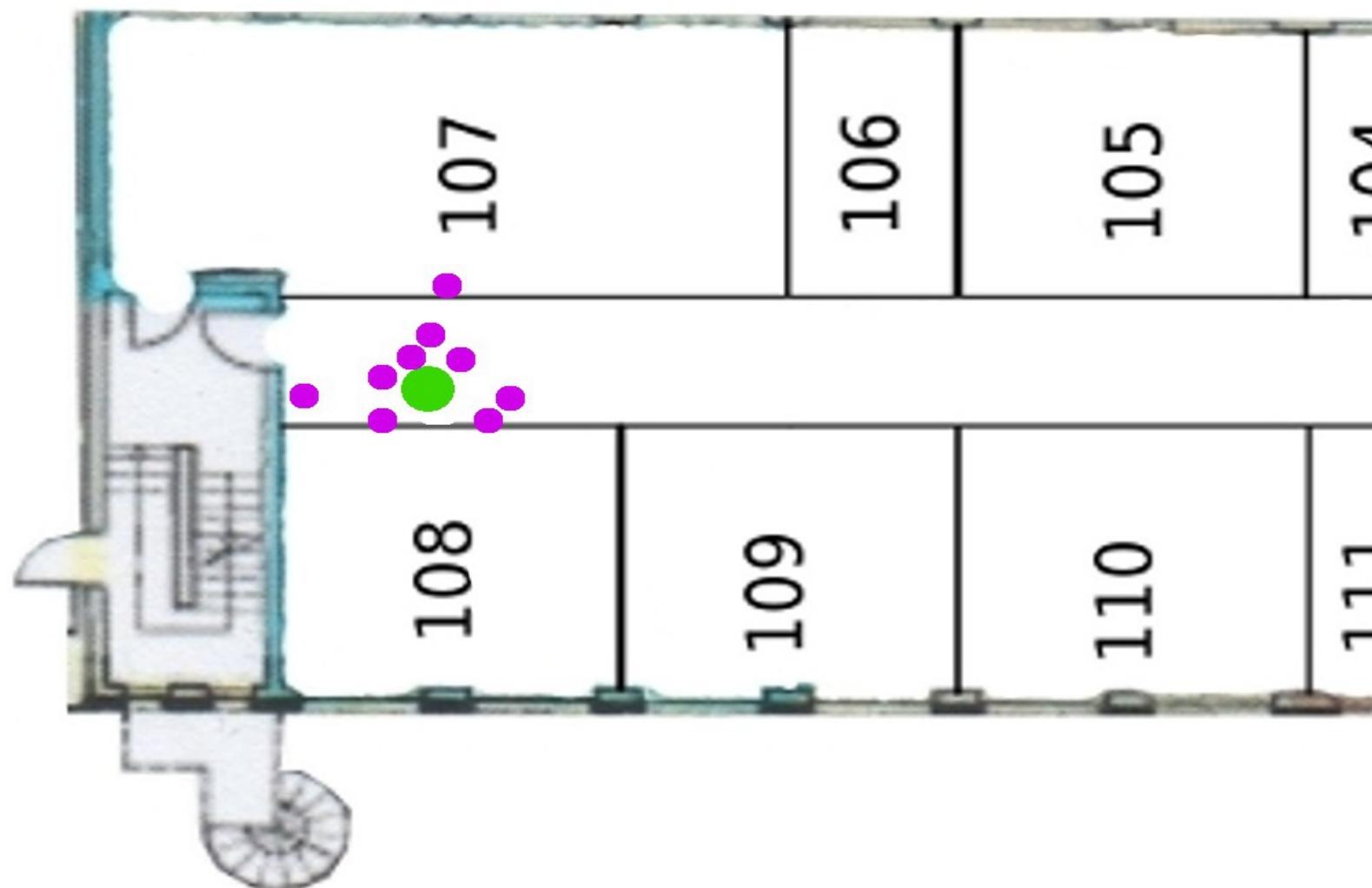
Partikel werden mit Rauschen in die durch das Dead Reckoning vorhergesagte Richtung verschoben



Partikel werden gemäß ihrer Distanz zur neu ermittelten WLAN-Position gewichtet

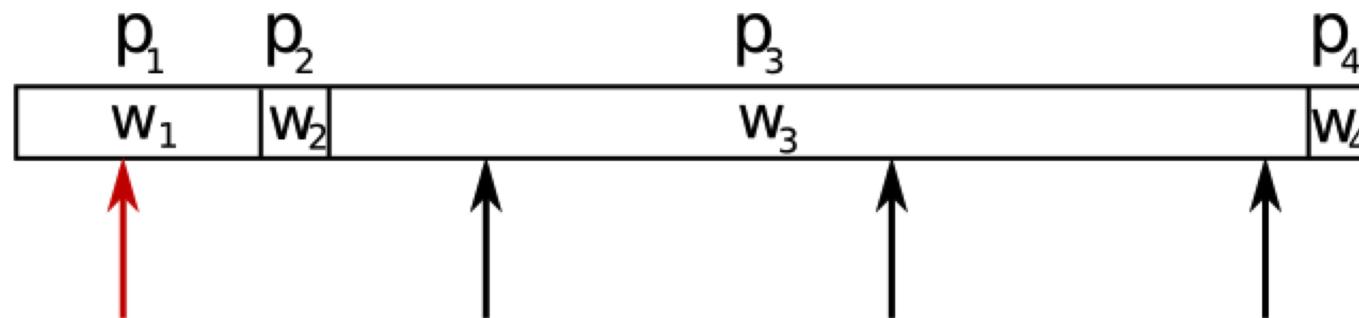


Neue Partikel werden entsprechend der Gewichtung erzeugt und die wahrscheinlichste Position daraus abgeleitet



## Stochastic Universal Sampling

- Häufigkeit der Partikel proportional zur Gewichtung
- Verteilungsfunktion wird im gleichen Intervall ( $1/\# \text{Partikel}$ ) abgetastet. Ergebnis: gesampelte Partikelmenge
- Gesampelter Partikelmenge wird Streuung (gaußverteilt) hinzugefügt



$$X = \{p_1, p_2, p_3, p_4\} \rightarrow X' = \{p_1, p_3, p_3, p_3\}$$

# Agenda

Wie machen wir Indoor Navigation?

Welche Komponenten verwenden wir?

Sensoren

WLAN

Bewegungssensoren

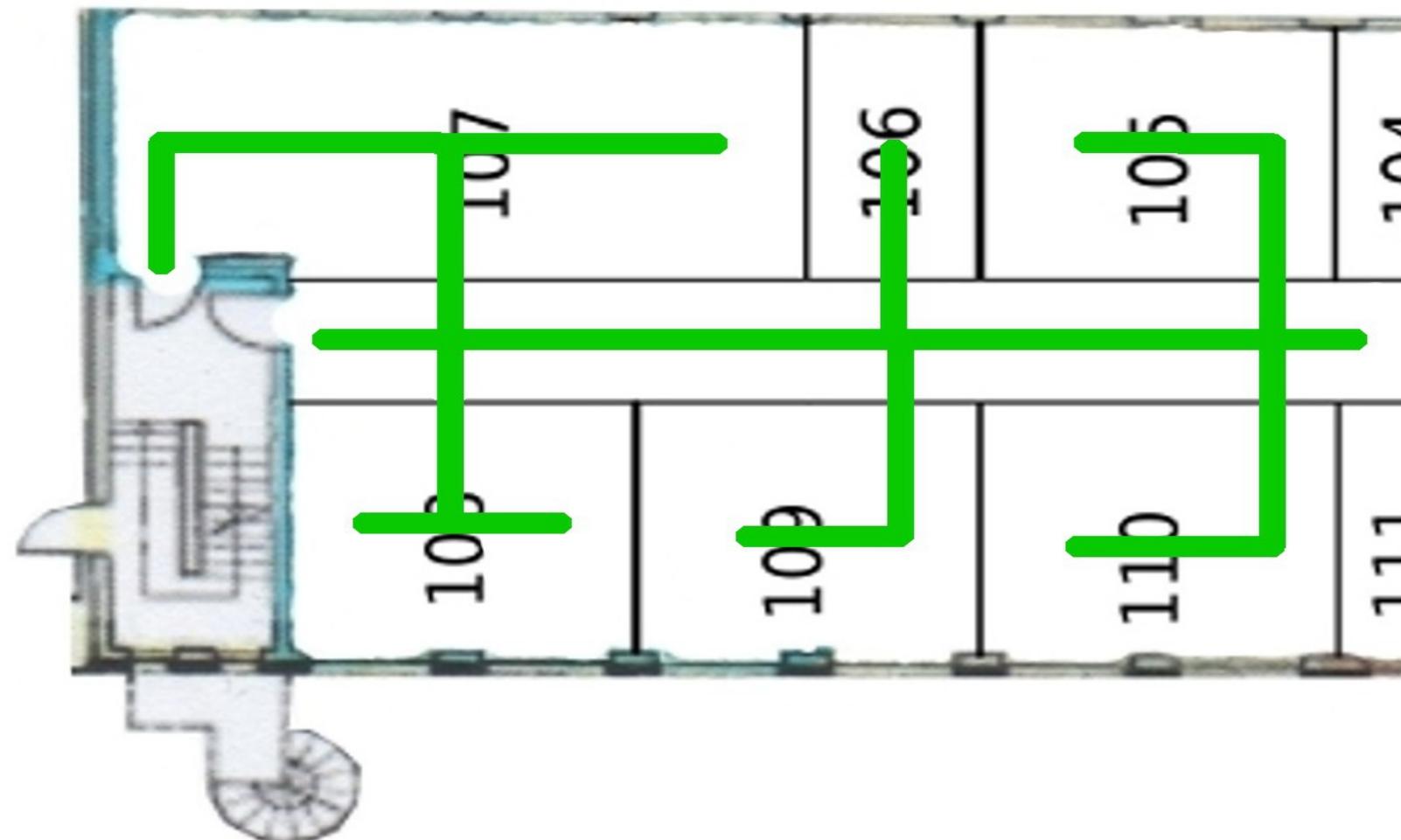
Fusion aller Sensoren



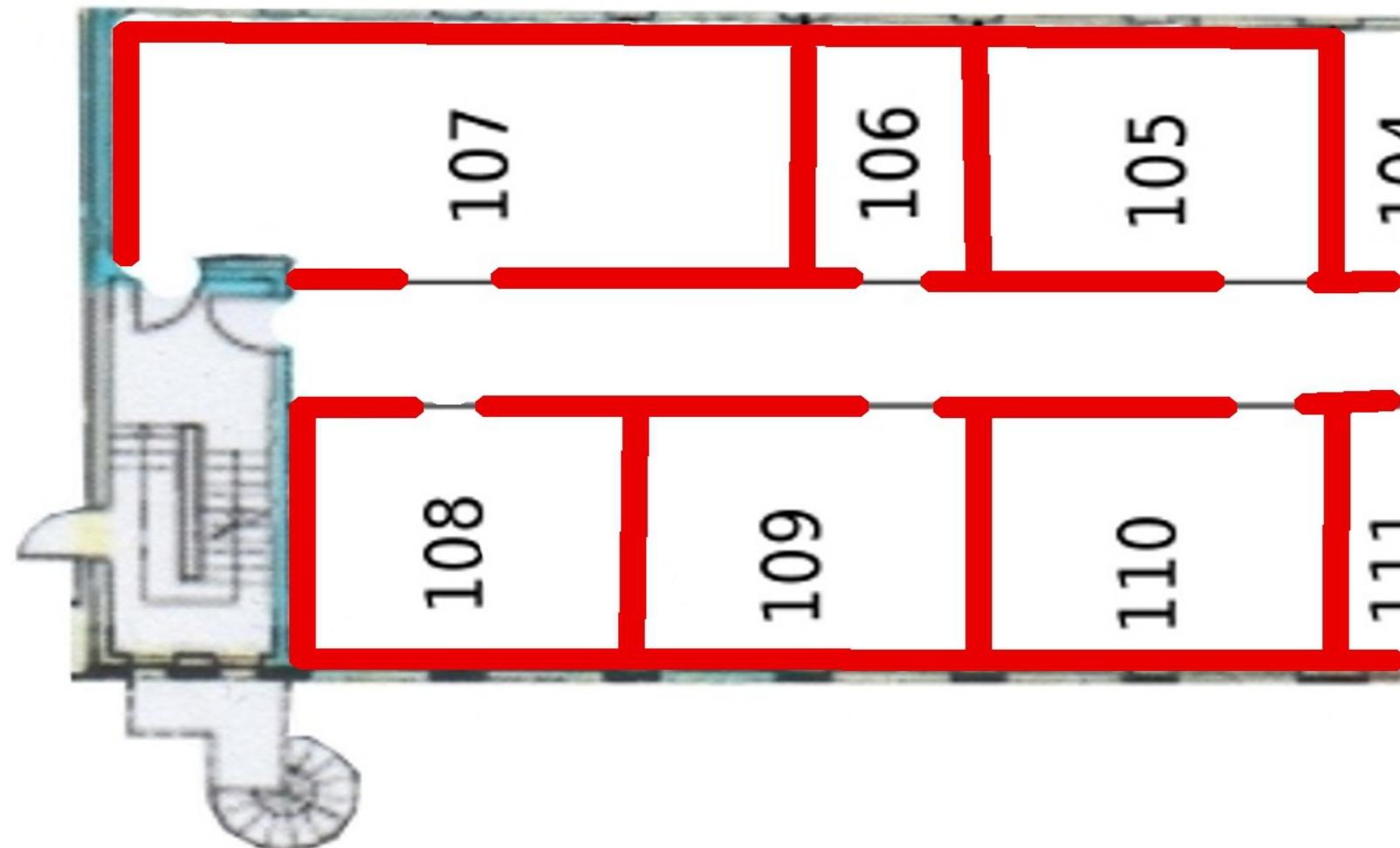
Matching auf Umgebungsinformationen

Ausblick & Demo

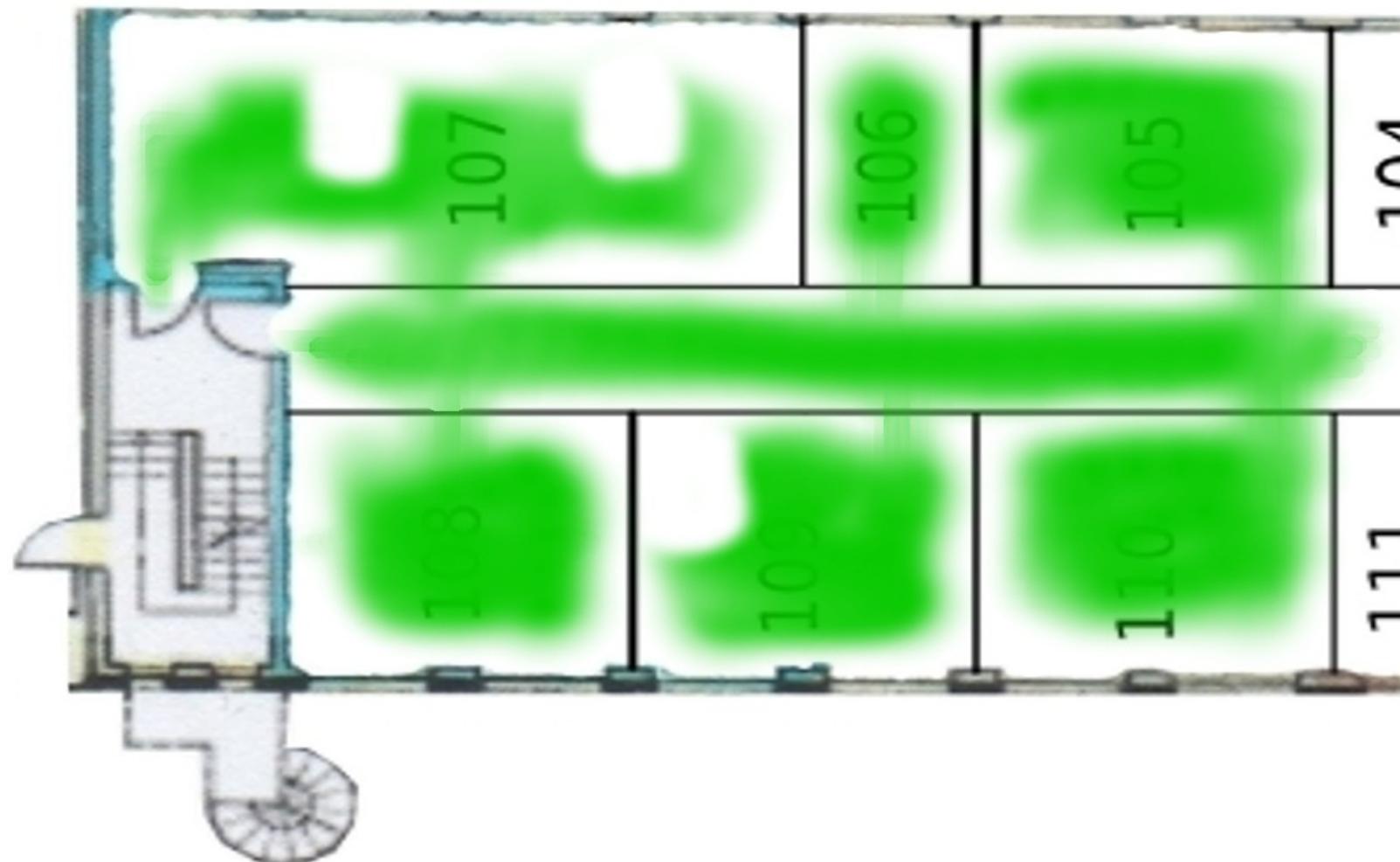
Lokalisierungsrelevante Informationen können in die Karte kodiert werden:  
(binär, positiv, statisch/dynamisch)



Lokalisierungsrelevante Informationen können in die Karte kodiert werden:  
(binär, negativ, statisch/dynamisch)



Lokalisierungsrelevante Informationen können in die Karte kodiert werden:  
(probabilistisch, statisch)



# Agenda

Wie machen wir Indoor Navigation?

Welche Komponenten verwenden wir?

Sensoren

WLAN

Bewegungssensoren

Fusion aller Sensoren

Matching auf Umgebungsinformationen



Ausblick & Demo

## Pro:

- Laufzeit ist grundsätzlich direkt proportional zur Entfernung

## Contra:

- Ohne freies Sichtfeld: Reflektionen, Brechungen
- Zeiten sind, dank Lichtgeschwindigkeit, sehr kurz, daher braucht man:
  - große Entfernungen oder
  - sehr genaue Uhren oder
  - sehr viele Messungen

**Time of Arrival** (TOA): **Wie lange** ist das Signal von Basisstationen zum Empfänger unterwegs (oder andersrum, oder auch hin und zurück)?

**Time Difference of Arrival** (TDOA): In **welchem Abstand** kommen synchronisierte Signale beim Empfänger an oder ein Signal des Empfängers bei den synchronisierten Basisstationen?

Zumindest TDOA benötigt spezielle Infrastruktur.



**Das ganze Team:**

Andreas Grau  
Andrei Boulgakov  
Atanas Alexandrov  
Désirée Amling  
Dino Omanovic  
Jens Maiero  
Jan Schmitt  
Martin Gernhardt  
Miluska Pech  
Nils Neumaier  
Ricarda Steffens  
Robert Linden  
Sven Schumann  
Thomas Kudla  
Timo Kanera

**Vielen Dank!**

# Kontakt



**Andreas Grau** Mail: [a.grau@tarent.de](mailto:a.grau@tarent.de)

Rochusstraße 2-4  
53123 Bonn

**Robert Linden** Mail: [r.linden@tarent.de](mailto:r.linden@tarent.de)

Telefon: +49 (0) 228 54 881 -0  
Telefax: +49 (0) 228 54 881 -235