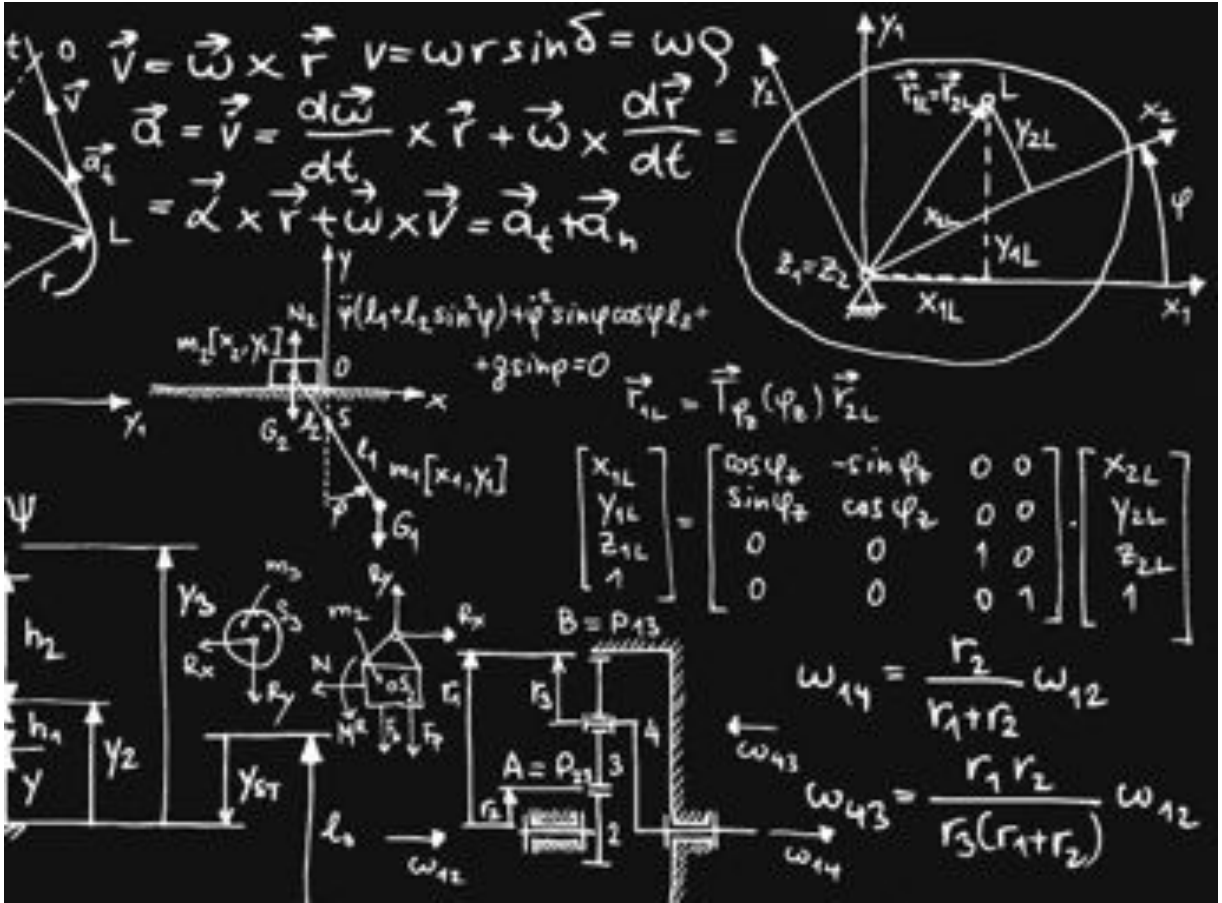
Data Meets Nice People II (Big Data Week Edition)

Users Know Best:
Data-driven Recommendations For Outdoor Activities

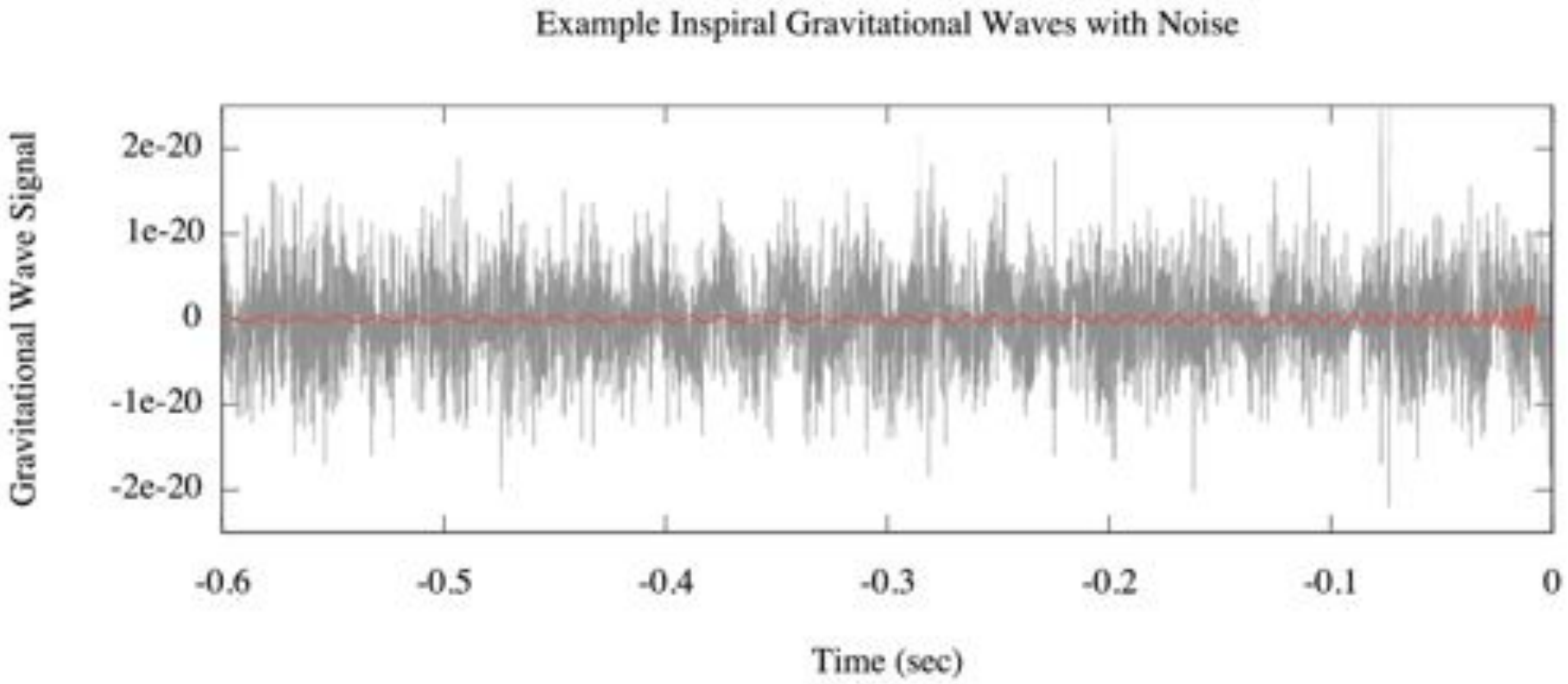
Lucía Santamaría
Recommender Engineer @ komoot

A few facts about… myself



A few facts about… myself

A few facts about… myself

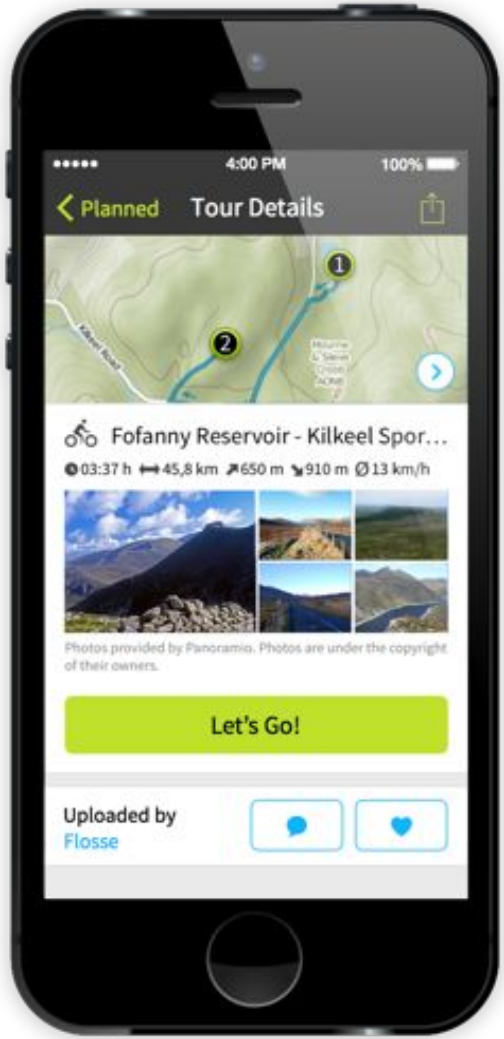


A few facts about… myself

A few facts about…

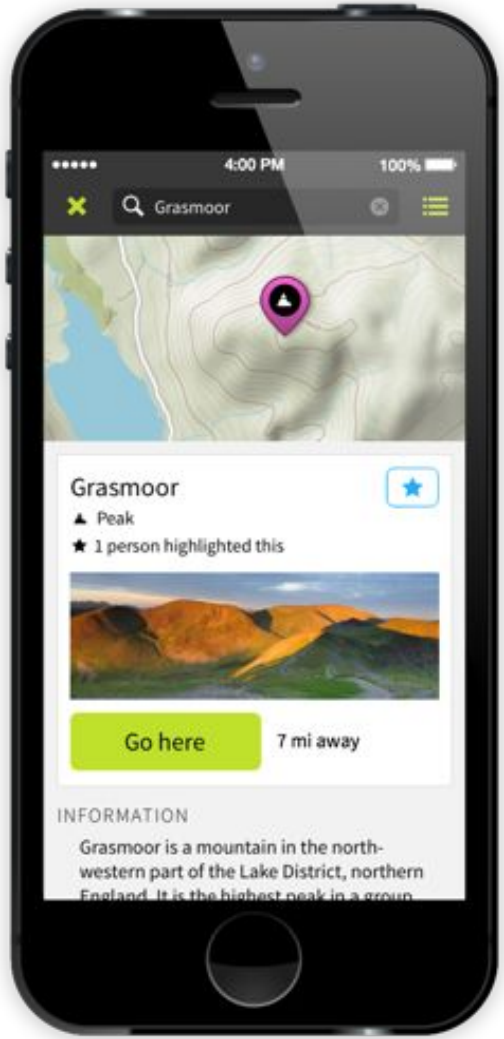
komoot is your personal outdoor guide

Great Outdoor Experiences,
with Simplicity, for Everyone

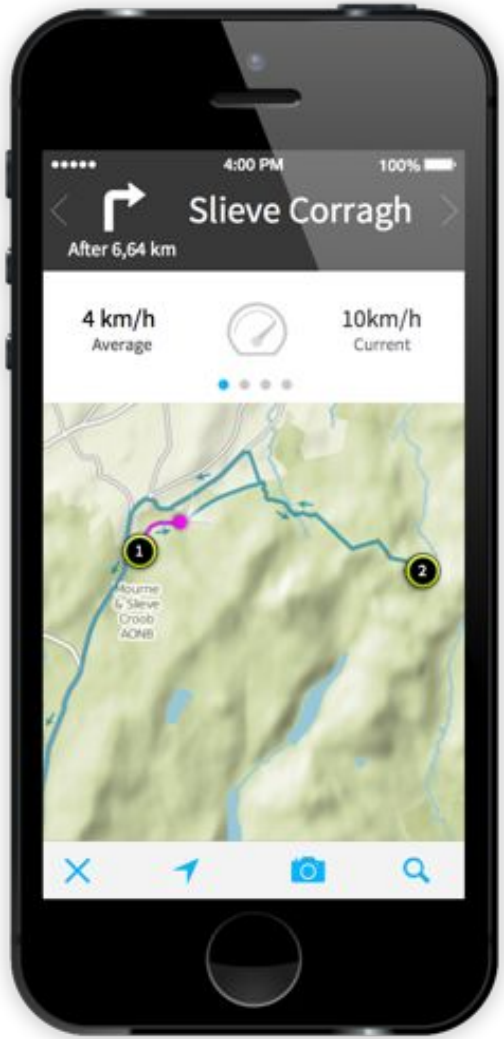


Route Planning.

Route Planning.



Start Exploring.

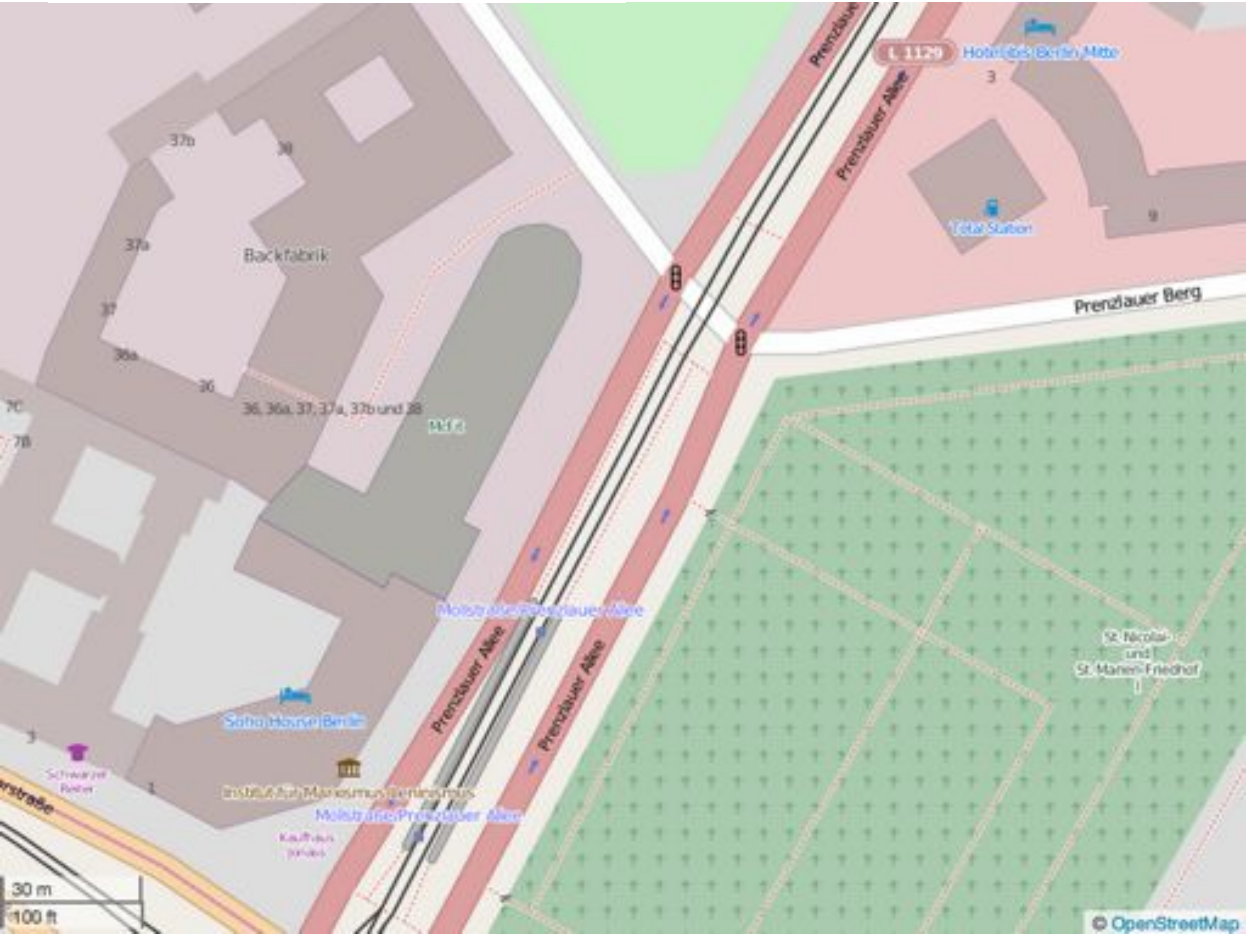


Voice Navigation.

Active Outdoors.

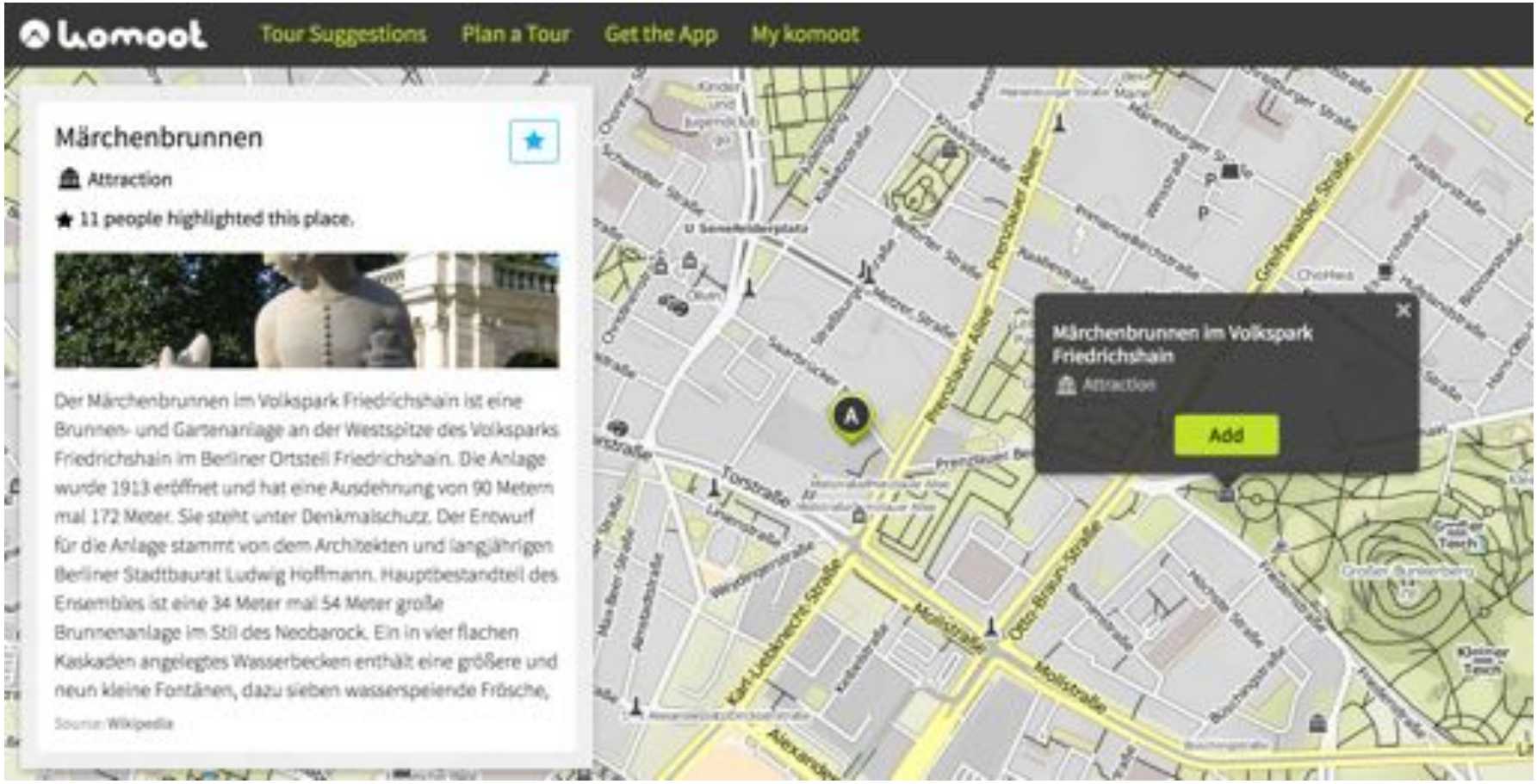
Recommending places to explore with komoot

OpenStreetMap Point Of Interest

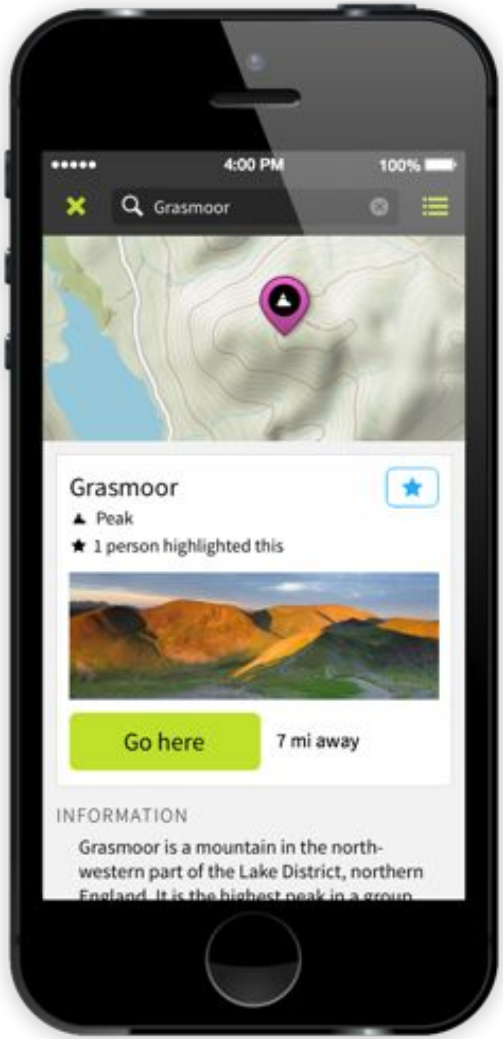


- Feature in a geodata set which occupies a point (opposed to linear features)
- Not necessarily very *interesting*
- **Categorized**
  - Churches, schools
  - Post offices, shops
  - Pubs
  - Car parks
  - Speed cameras
  - Tourist attractions
  - Nature: lakes, peaks, forests, parks, fountains

komoot Users can plan tours to OSM POIs in web

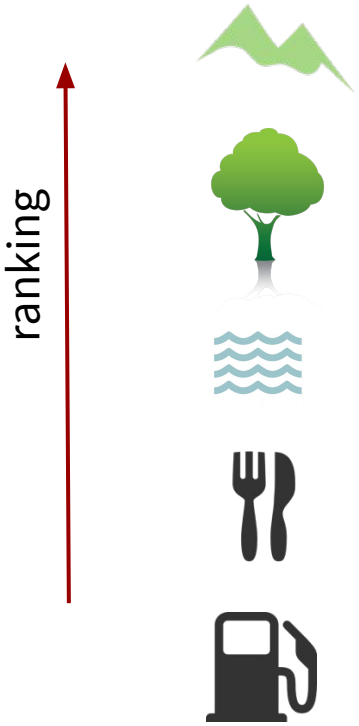


… and mobile clients



Sort POIs in map according to **static ranking of categories**

and present the N first results to the the User



Pure category ranking

```
...
ORDER BY ranking DESC;
```

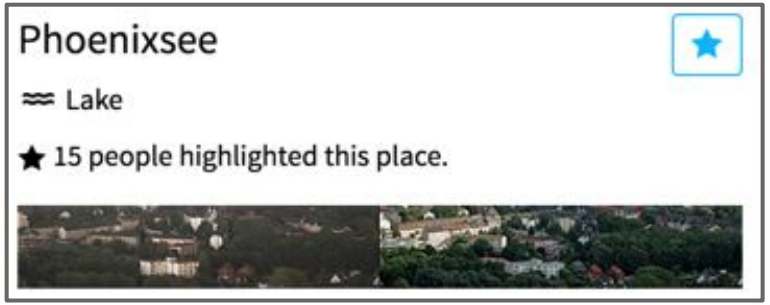
With penalization for repeated category

```
...
ORDER BY
    ranking
    - c1 *
    f(row_number() over
        (partition by category
        order by ranking DESC))
DESC;
```

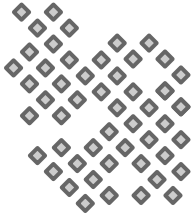
- 1st item of each category penalized **- c1 * f(1)**
- 2nd item penalized **- c1 * f(2)**
- …
- REAL ranking = ranking - penalization

[f(row) is a monotonic function]

Users Know Best



Refine rankings with user feedback → Users Know!



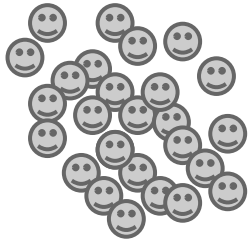
geo db
postgis

→

re-ranking algorithm

tours
mysql

messages
s3



Stack

PostGIS Spatial PostgreSQL

python™

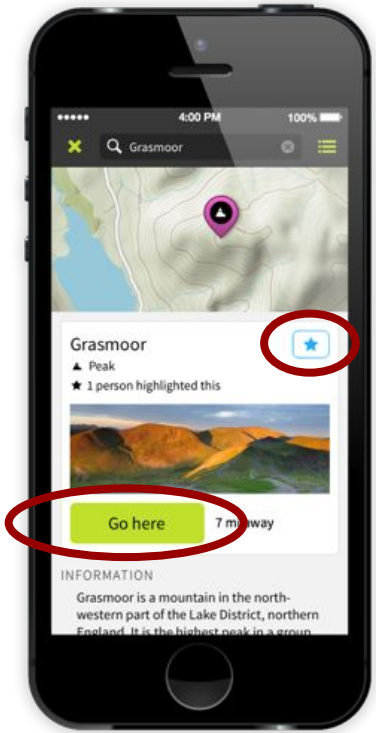
django

Leaflet

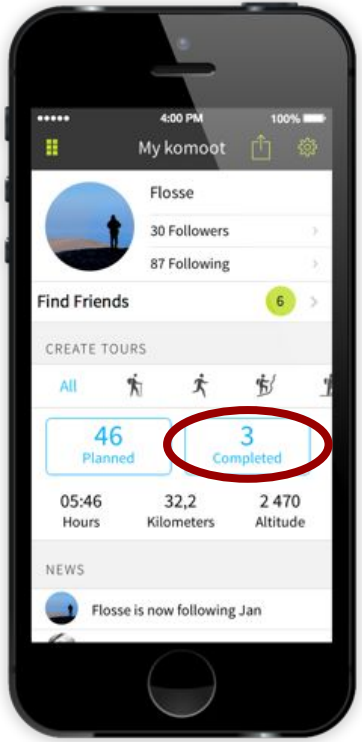
jQuery

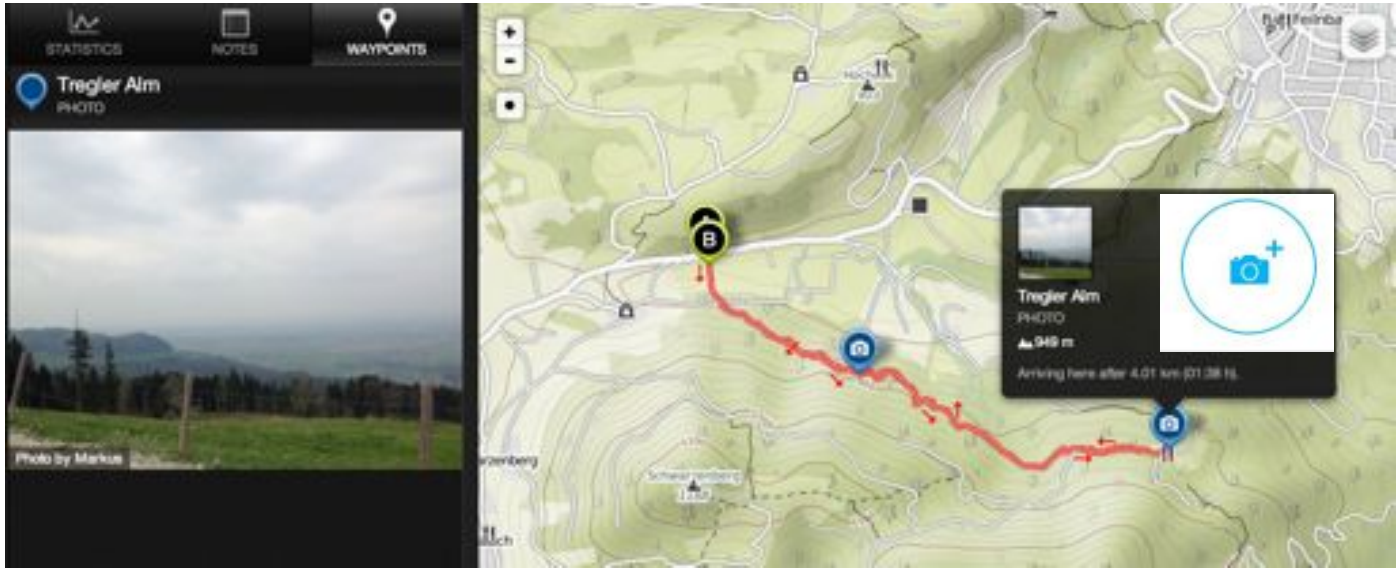
amazon webservices™

User feedback comes in many sizes and colors



User feedback comes in many sizes and colors





Dynamic, self-learning correction to rankings

static ranking

photos

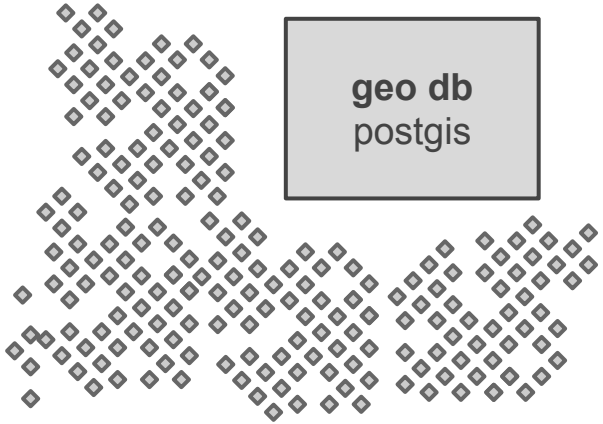
highlights

visits

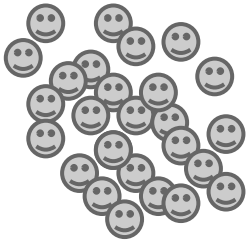
breaks

planned tours
vs.
POI views

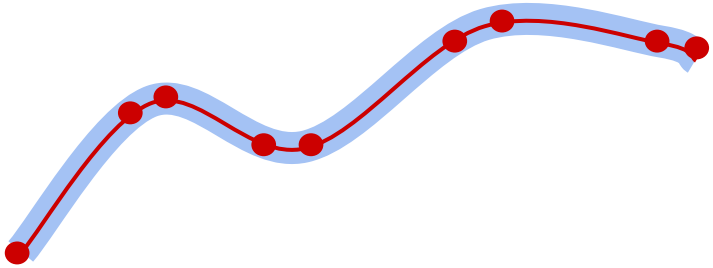
Sounds reasonable…
but how do we actually match events/messages to POIs?



~ 5 x 10^6 POIs



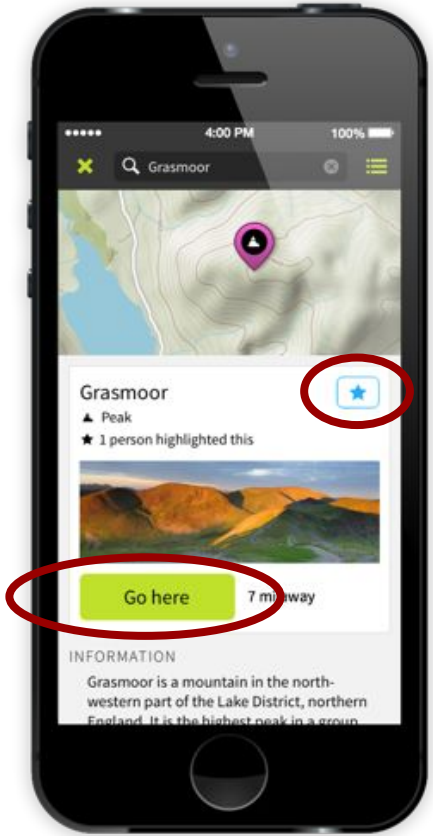
~ 10,000 tours / day



~ 1,000 nodes / tour
(LineString)

~ 10^{12} (1 million x 1 million)
operations / day

[1 day = 86400 ~ 10^5 sec]



HIGHLIGHTED PLACES

poi_id date username category

PLANNED TOUR TO POI

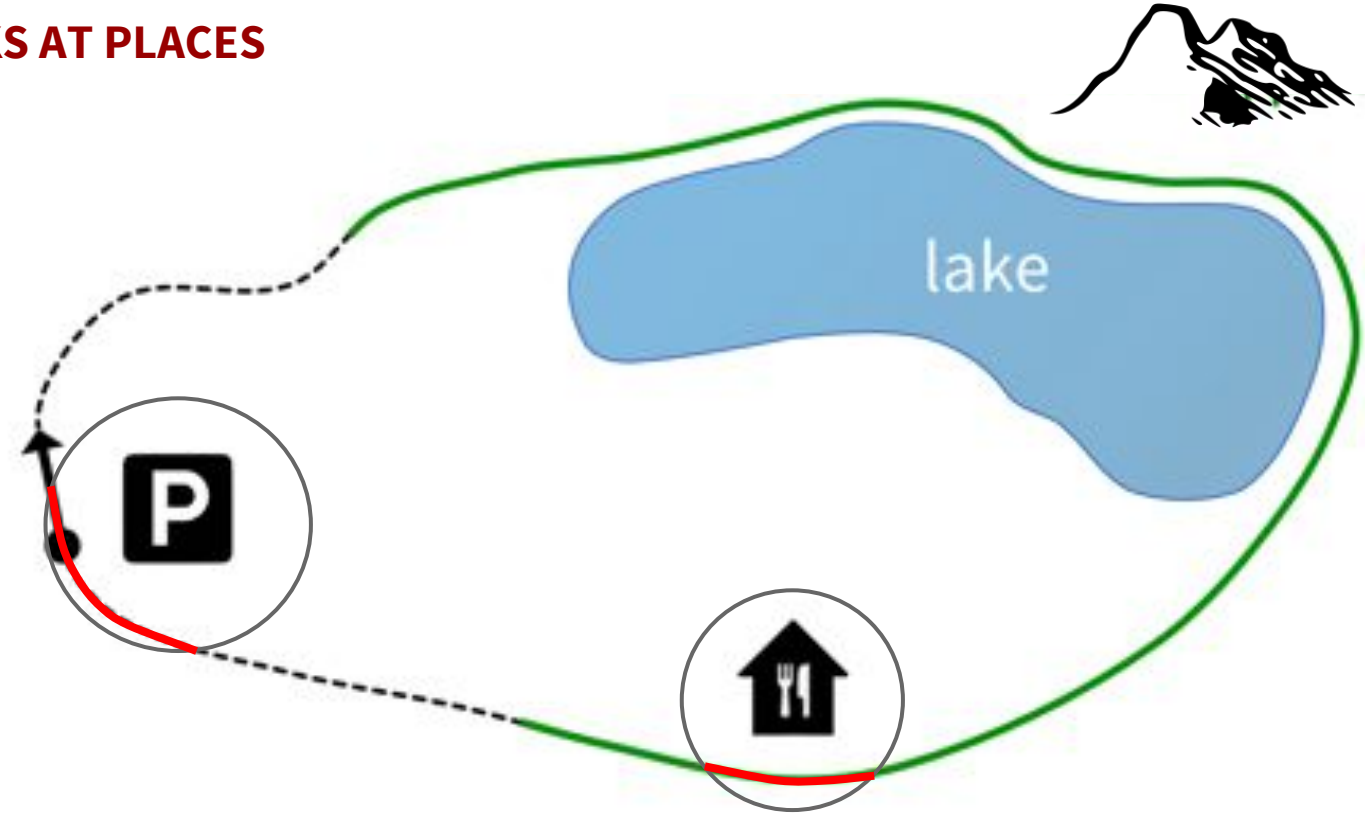
poi_id date username sport category

```
SELECT      poi_id, count(DISTINCT username)
FROM        planned_tour
GROUP BY    poi_id
```

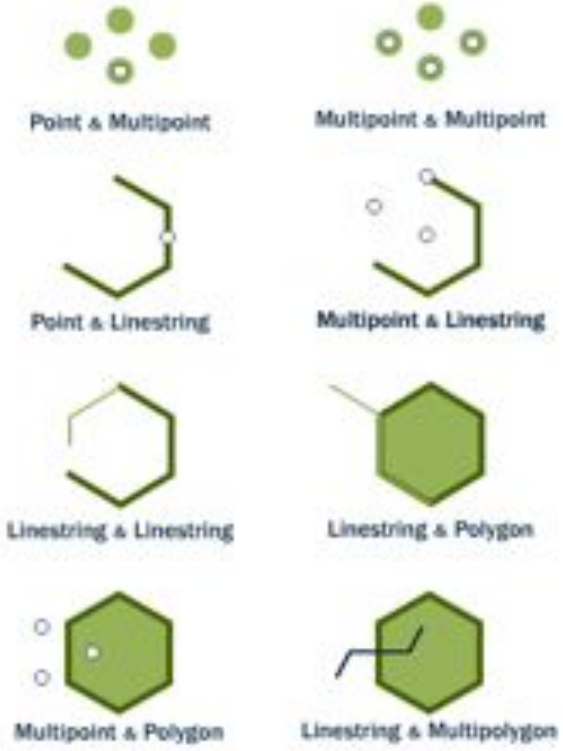
VISITED PLACES

PHOTOS AT PLACES

BREAKS AT PLACES



Intersects




```
SELECT name, ST_AsText(geom)
FROM nyc_subway_stations
WHERE name = 'Broad St';
```

```
POINT(583571 4506714)
```

```
SELECT name, boroname
FROM nyc_neighborhoods
WHERE ST_Intersects(geom, ST_GeomFromText('POINT(583571 4506714)',26918));
```

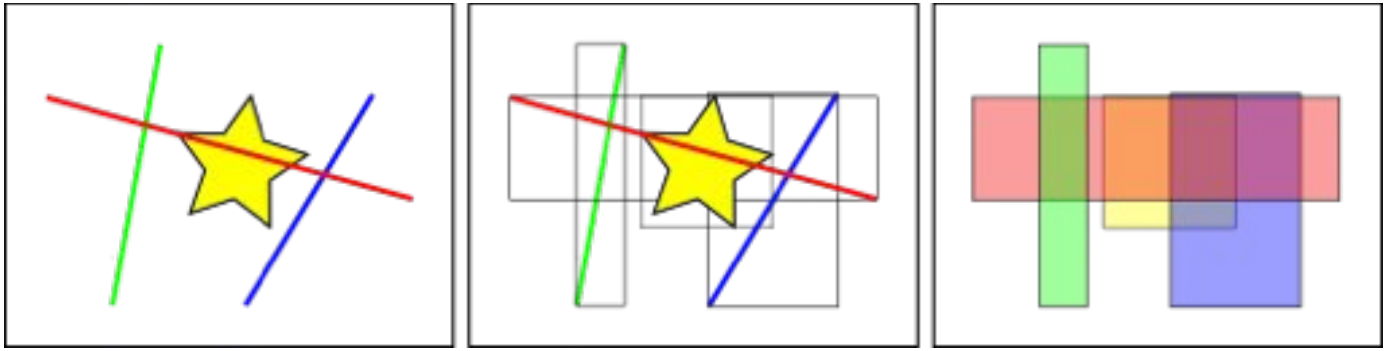
```
        name          | boroname
----------------------+-----------
 Financial District   | Manhattan
```

Very nice introduction to PostGIS:

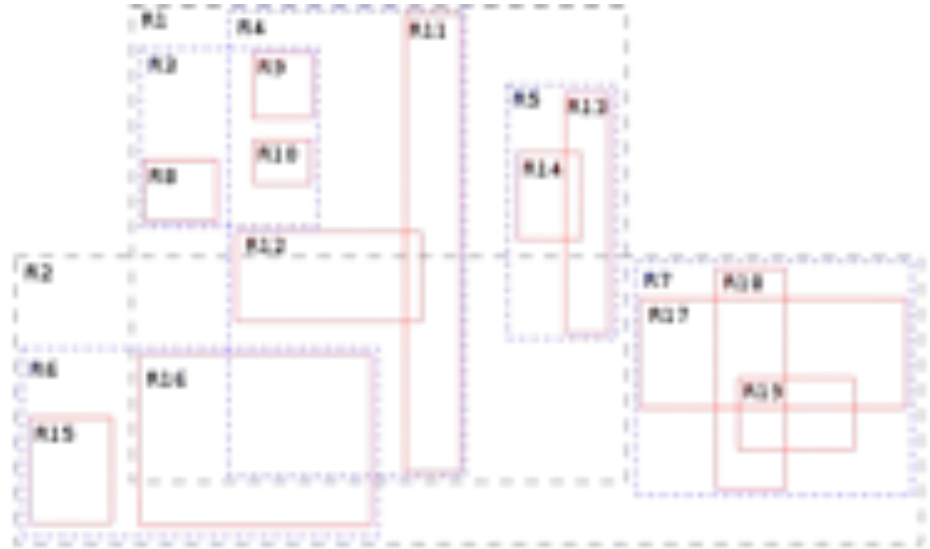
http://workshops.boundlessgeo.com/postgis-intro/

A word on **indices**

- Without indexing, any search for a feature would require a "sequential scan" of every record in the database.
- Standard database indexes create a hierarchical tree based on the values of the column being indexed.
- Spatial indexes are a little different – they are unable to index the geometric features themselves and instead index the bounding boxes of the features.

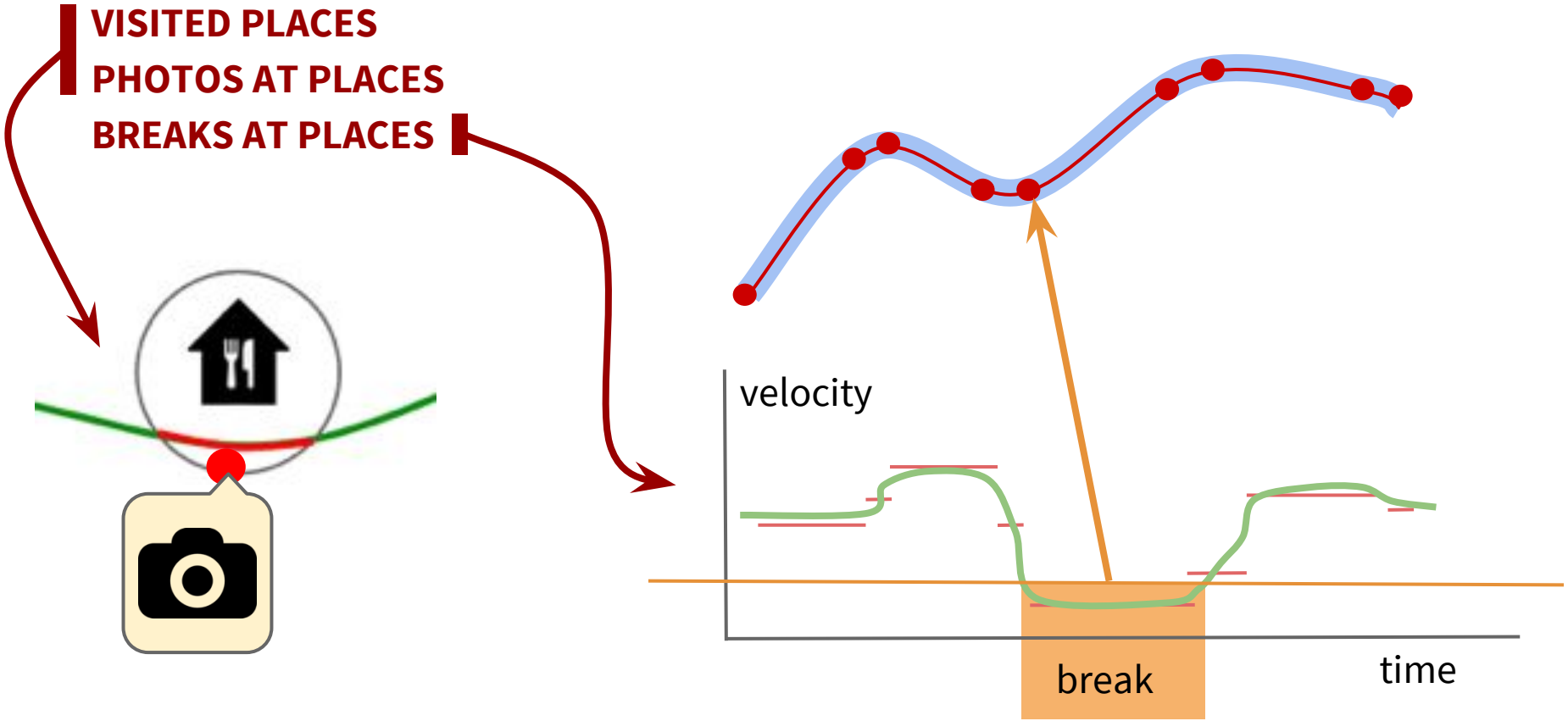


R-Tree Hierarchy



- "R" for Rectangle
- Group nearby objects and represent them with their minimum bounding rectangle in the next higher level of the tree
- A query that does not intersect the bounding rectangle also cannot intersect any of the contained objects
- For us: **20 min → 20 ms pro tour**
  [intersection of 5 x 10^6 POIs with 1000 nodes pro tour]

Using an in-memory r-tree object in django for POIs

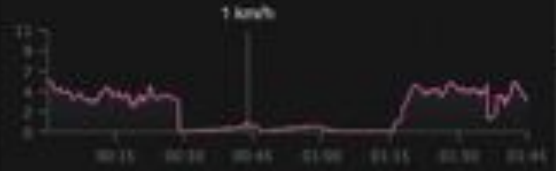

VISITED PLACES
PHOTOS AT PLACES
BREAKS AT PLACES
velocity
time
break

Sonna Alp

Made by Markus on 03/10/13
with the komoot App

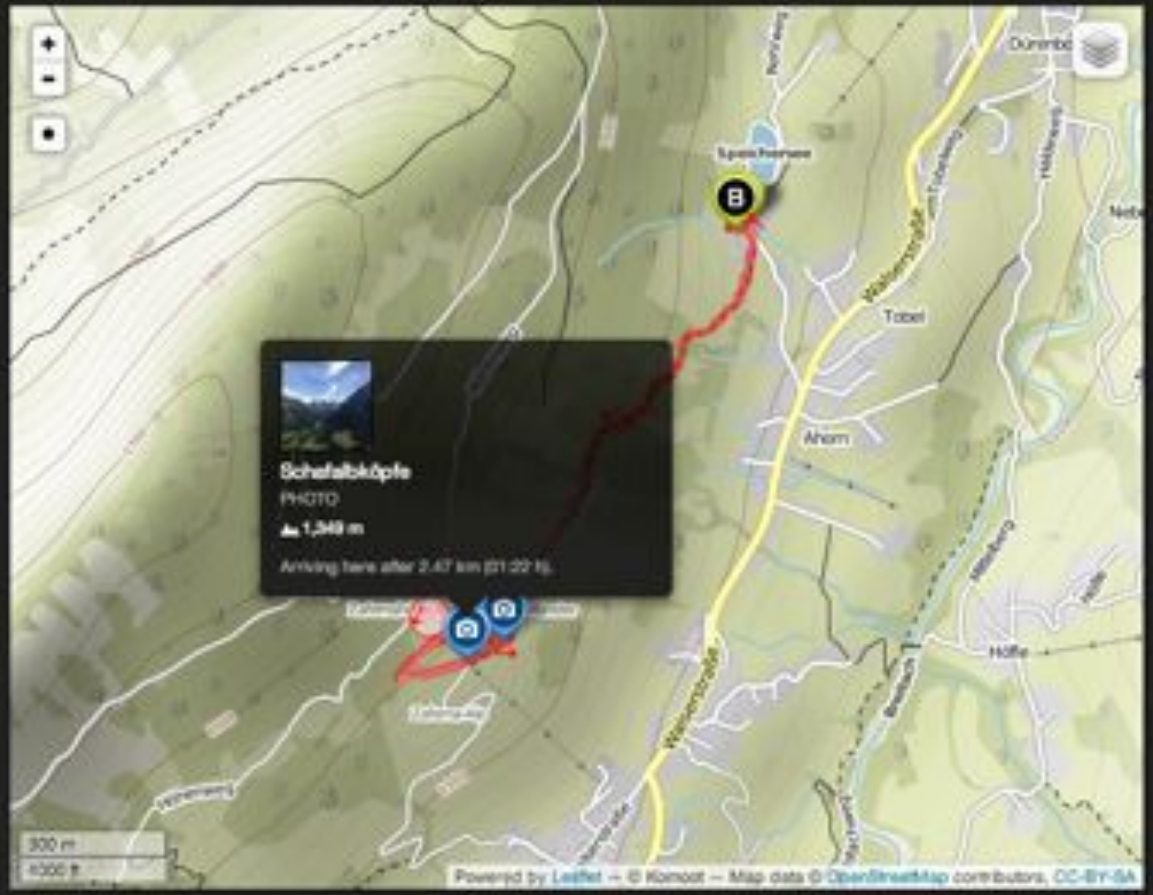
HIKING	3.9 km	01:43 h	210 m	190 m
	DISTANCE	DURATION	↑ UPHILL	↓ DOWNHILL

STATISTICS | NOTES | WAYPOINTS



Highest Point	1,420 m
Lowest Point	1,220 m
Speed	2 km/h

Tweet | Like 1



Dynamic, self-learning correction to rankings

static ranking

photos

highlights

visits

breaks

planned tours
vs.
POI views

lucia@komoot.de