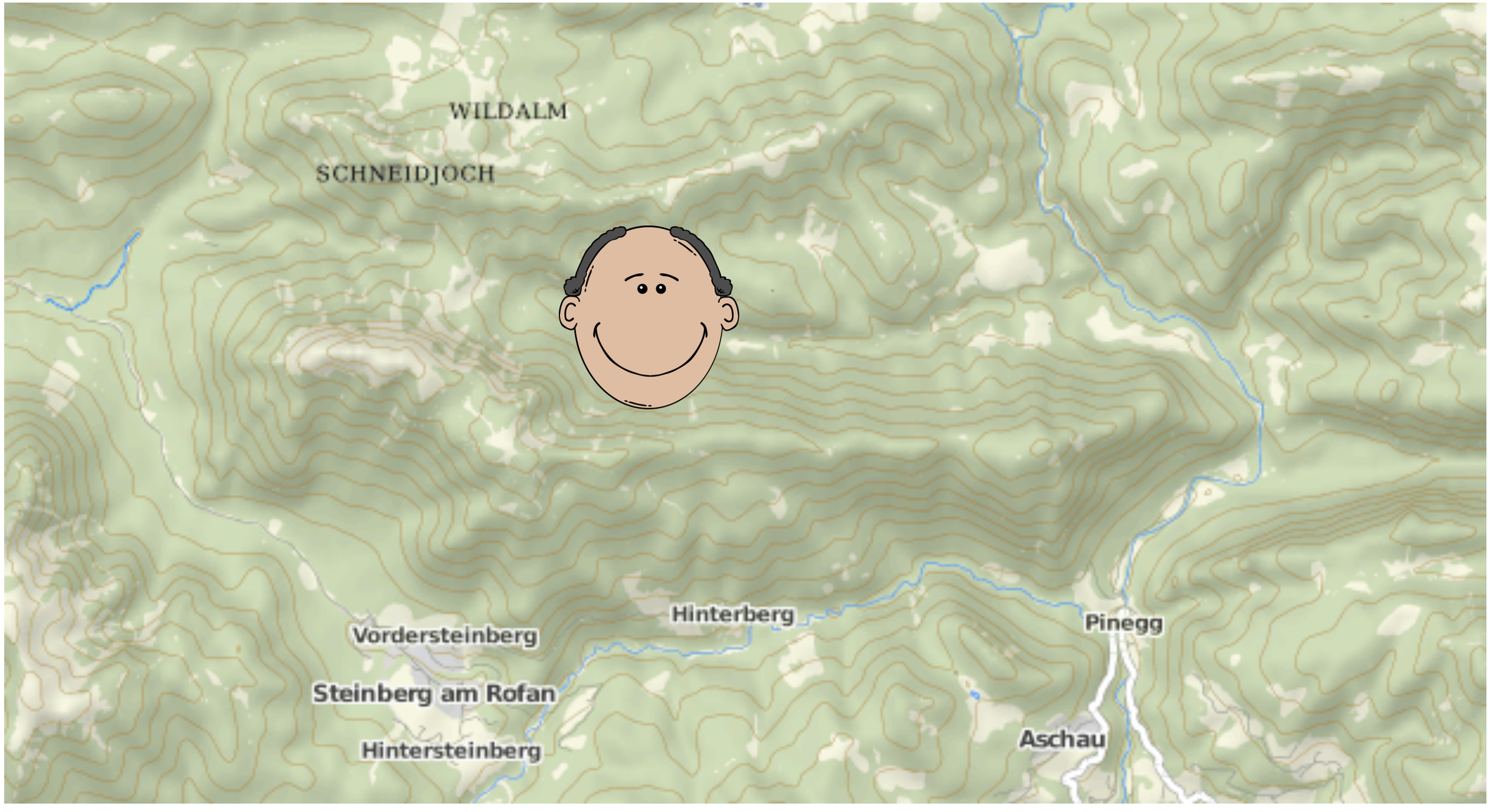
A Preference SQL Approach to Improve Context-Adaptive Location-Based Services for Outdoor Activities

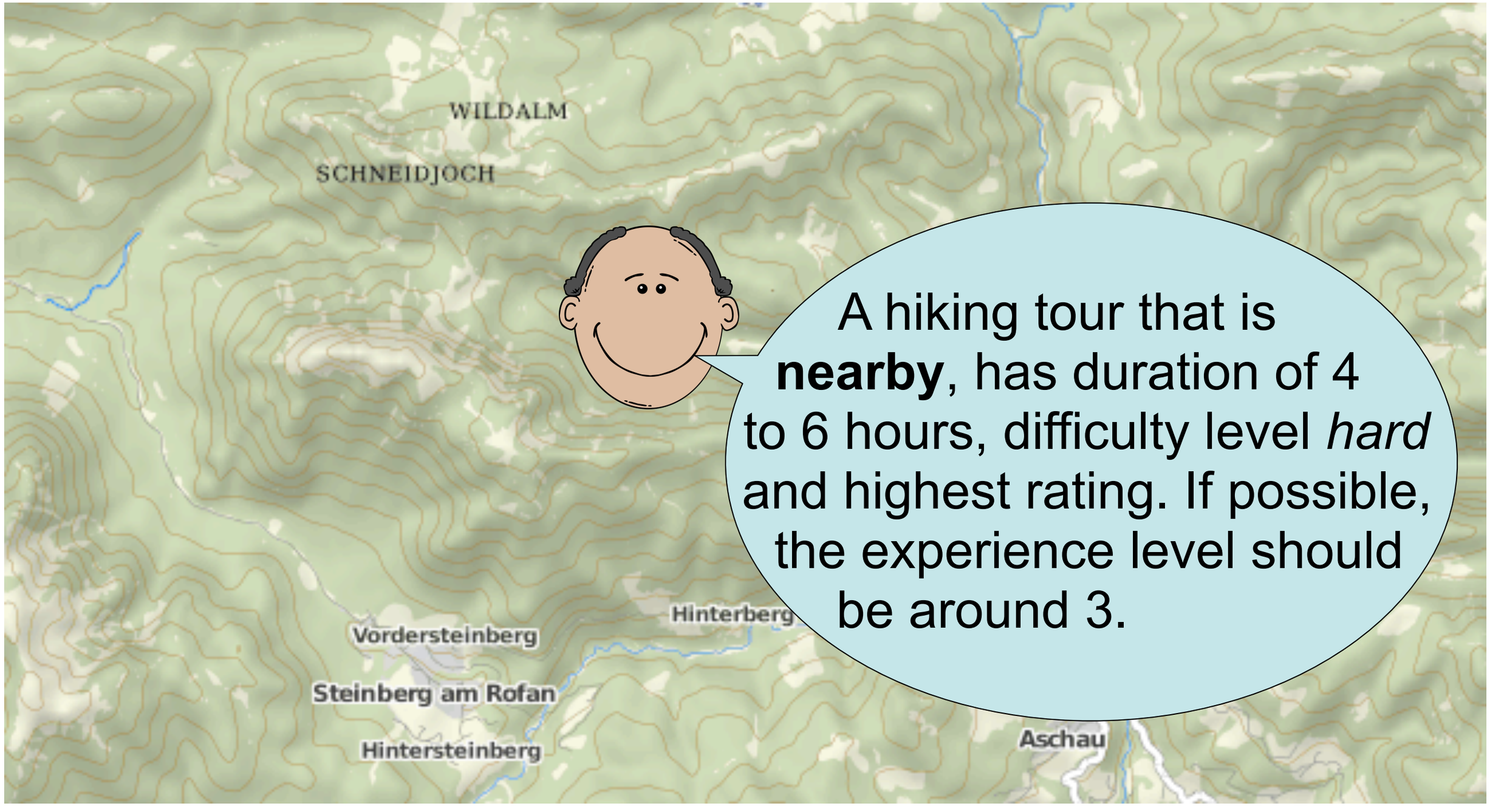
F. Wenzel, M. Soutschek, W. Kießling

Vienna, 21st Nov. 2011

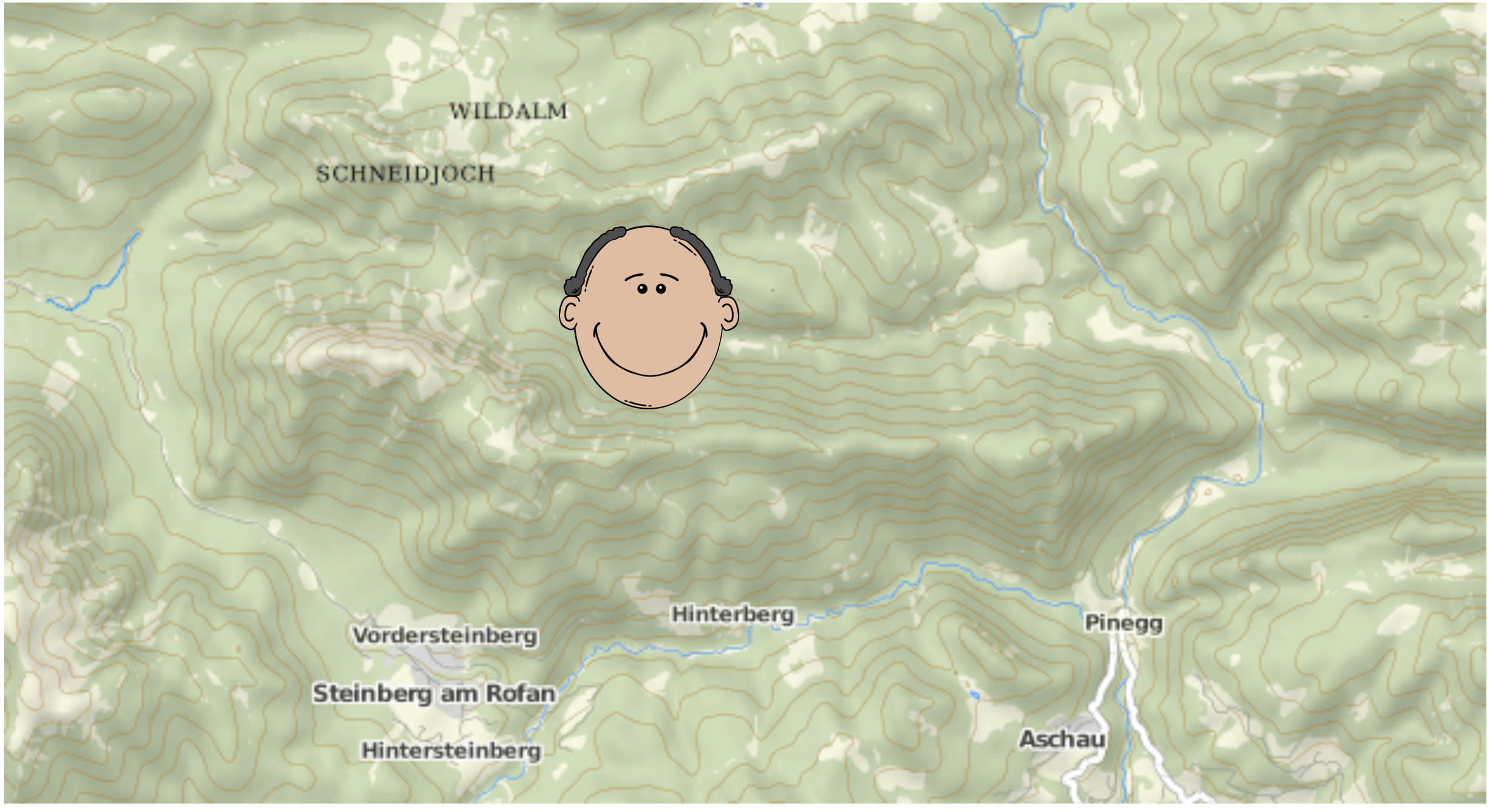
Running Example



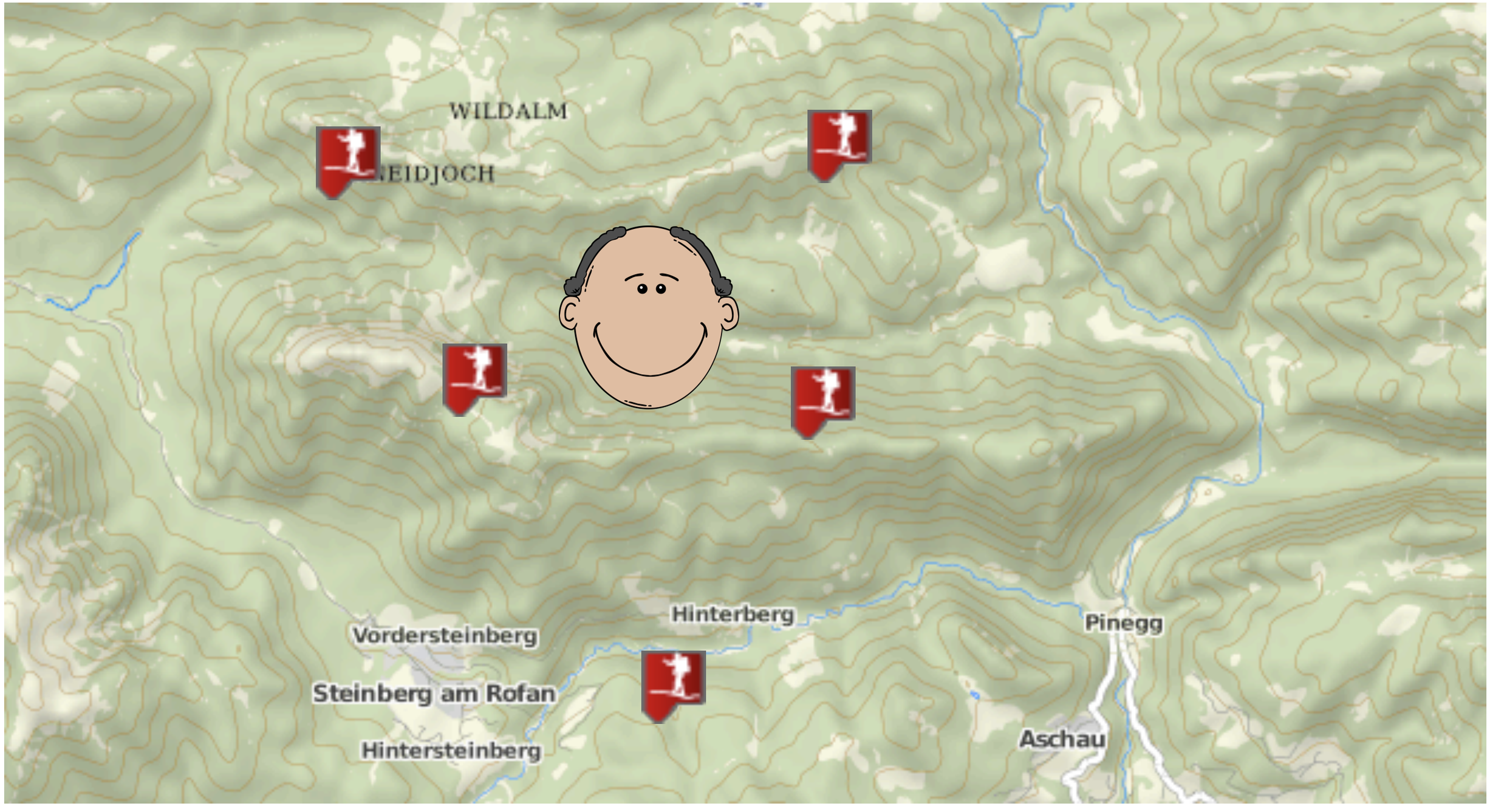
Running Example



Running Example

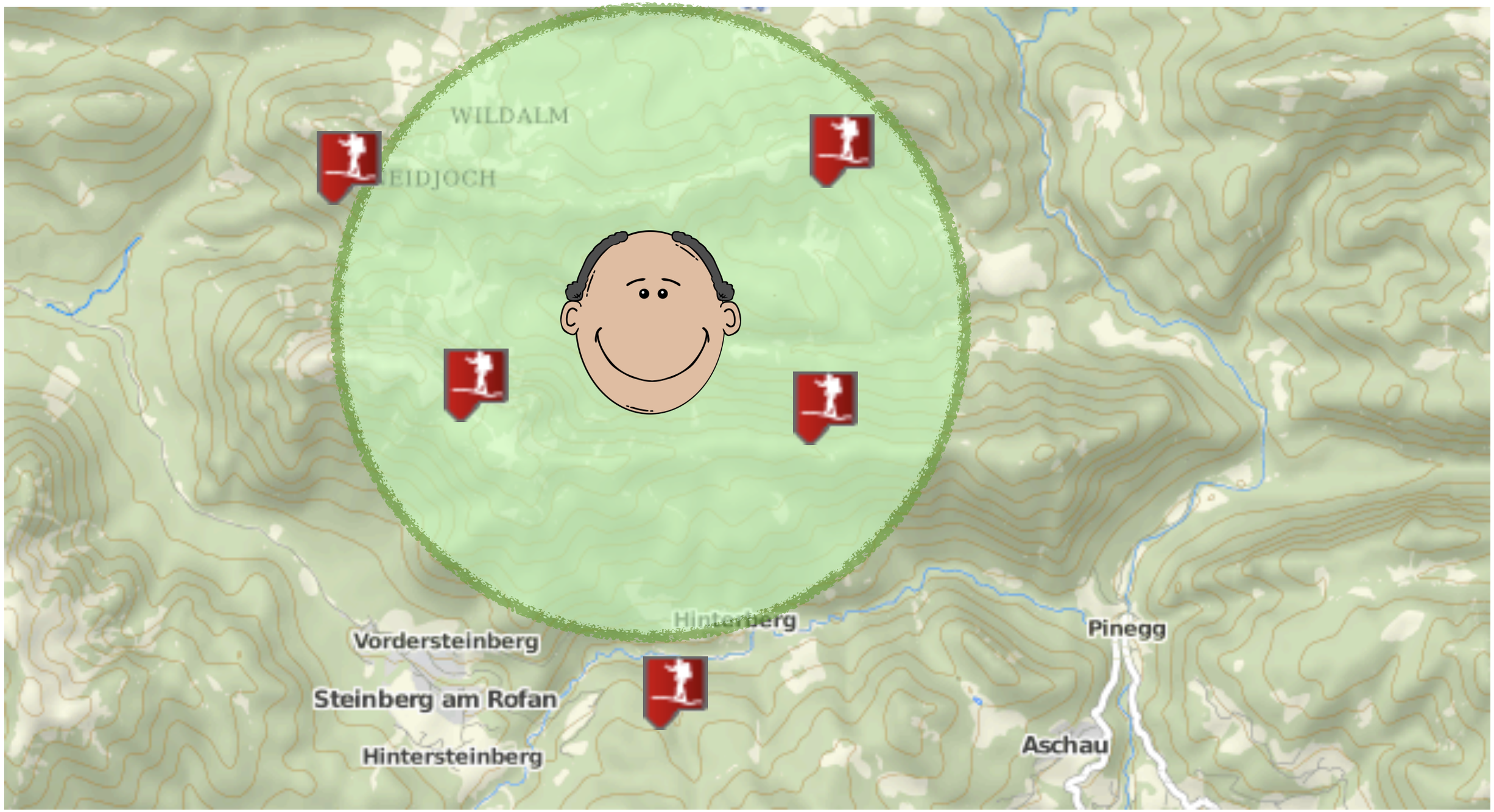


Running Example

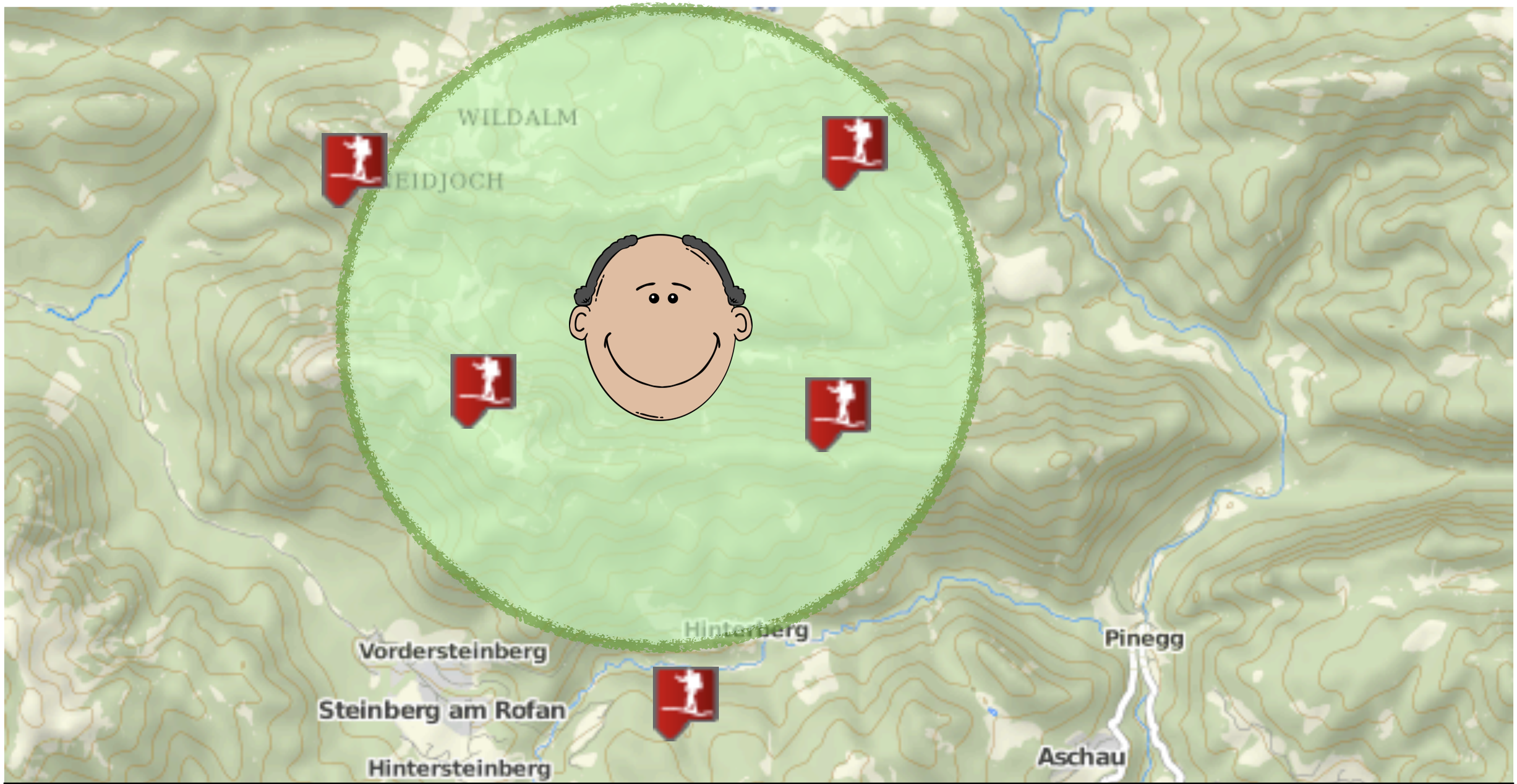


Running Example

Universität Augsburg University

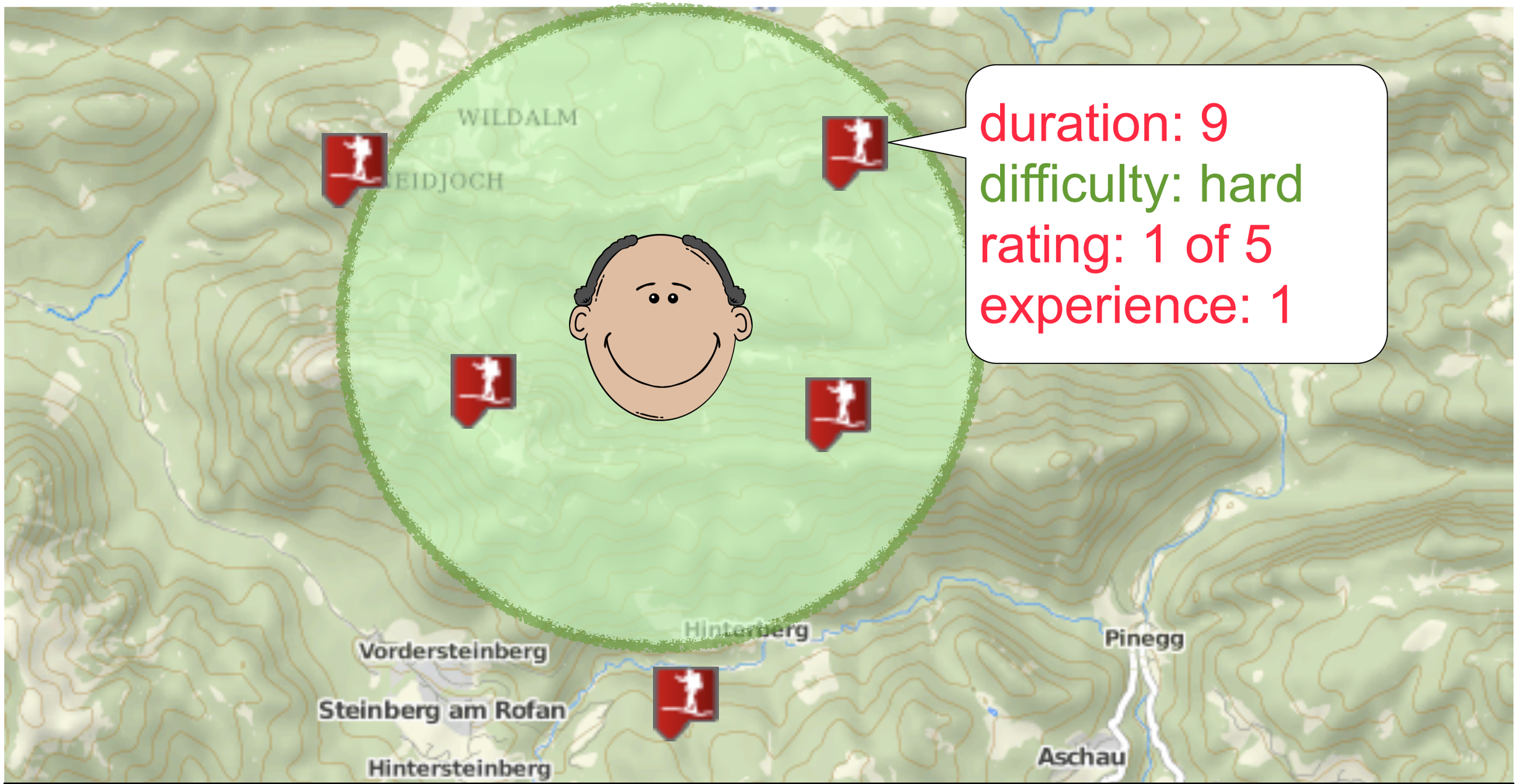


Running Example



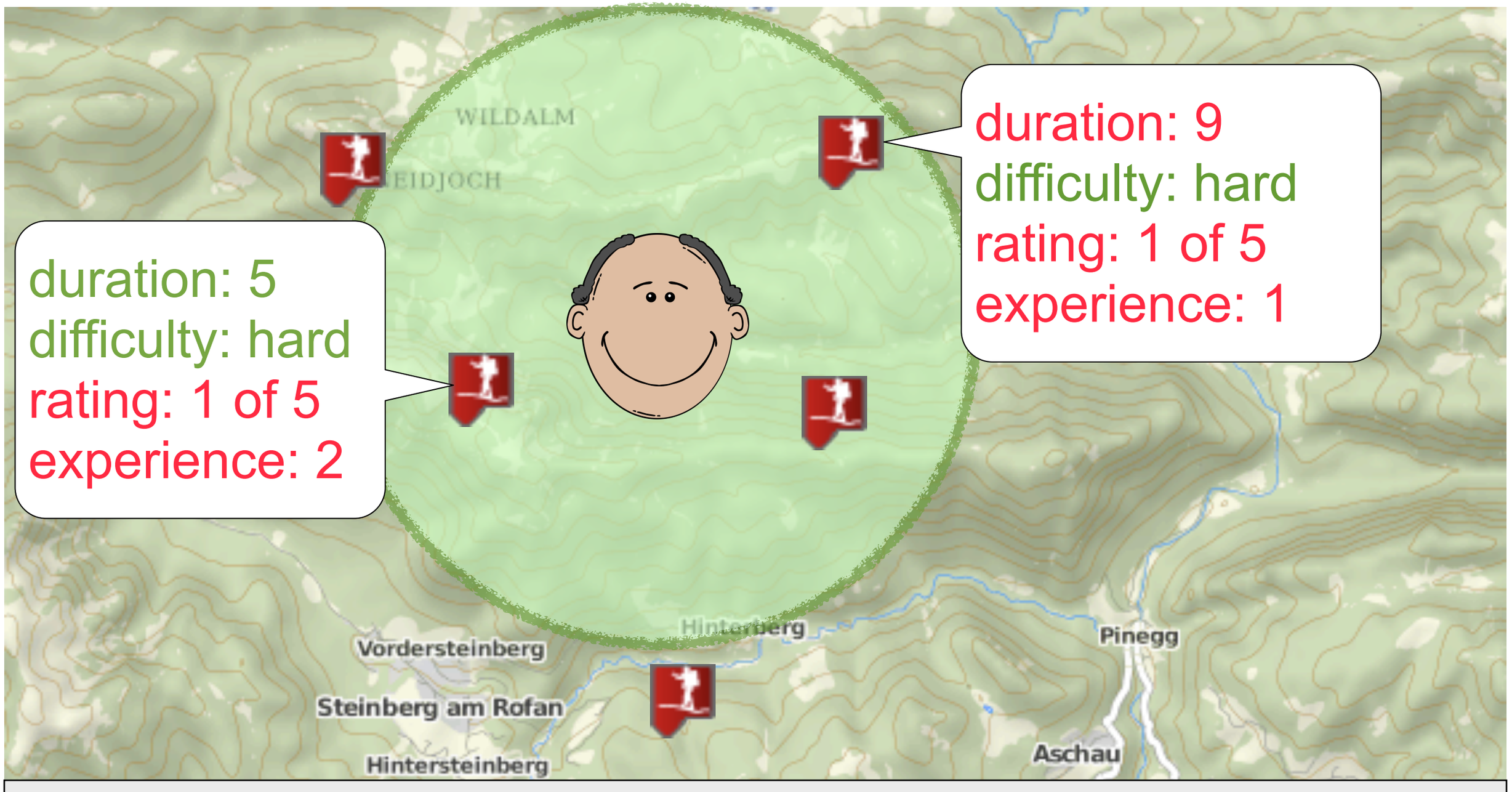
hiking tour: nearby, duration 4 to 6 hours, difficulty *hard*, highest rating, experience level around 3.

Running Example



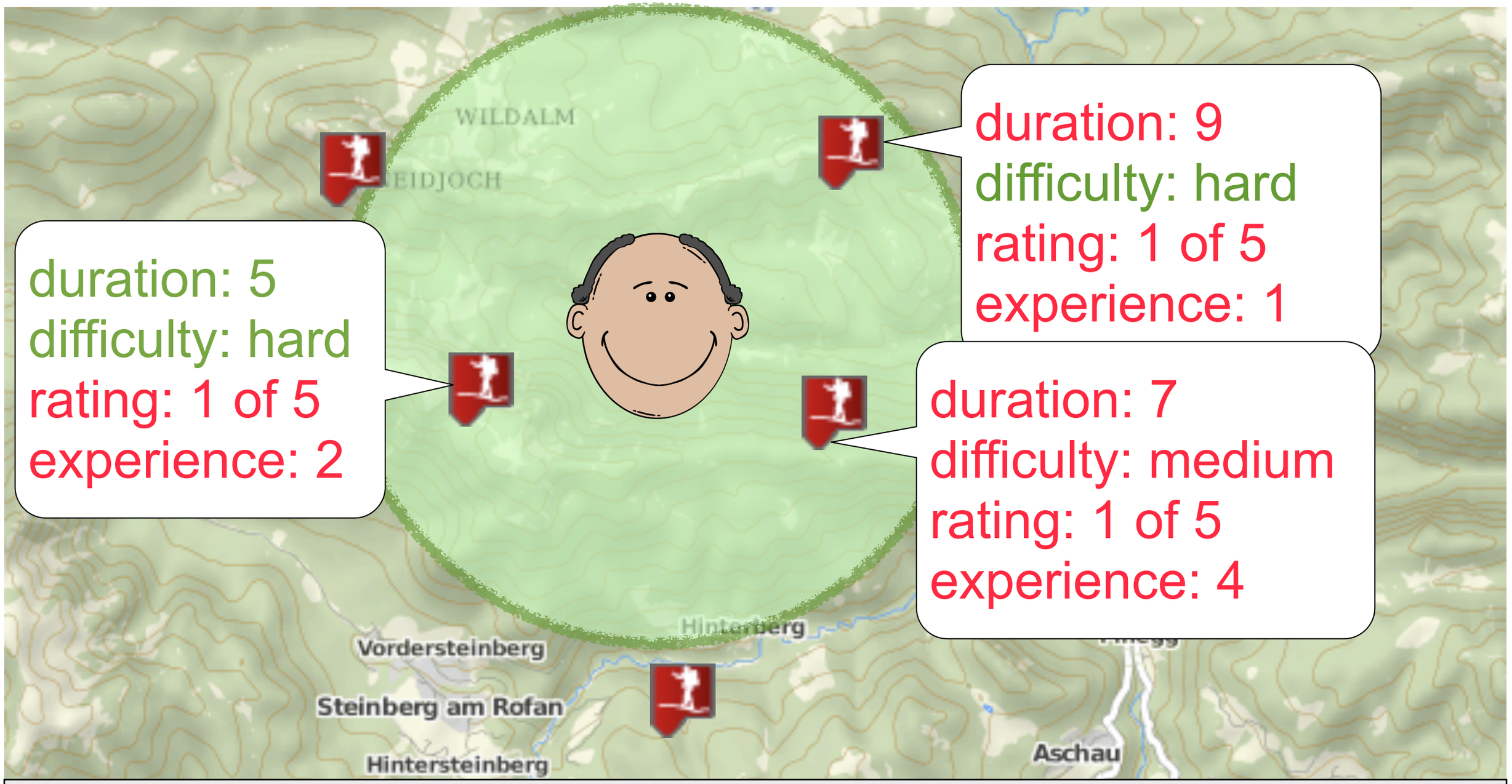
hiking tour: nearby, duration 4 to 6 hours, difficulty *hard*, highest rating, experience level around 3.

Running Example



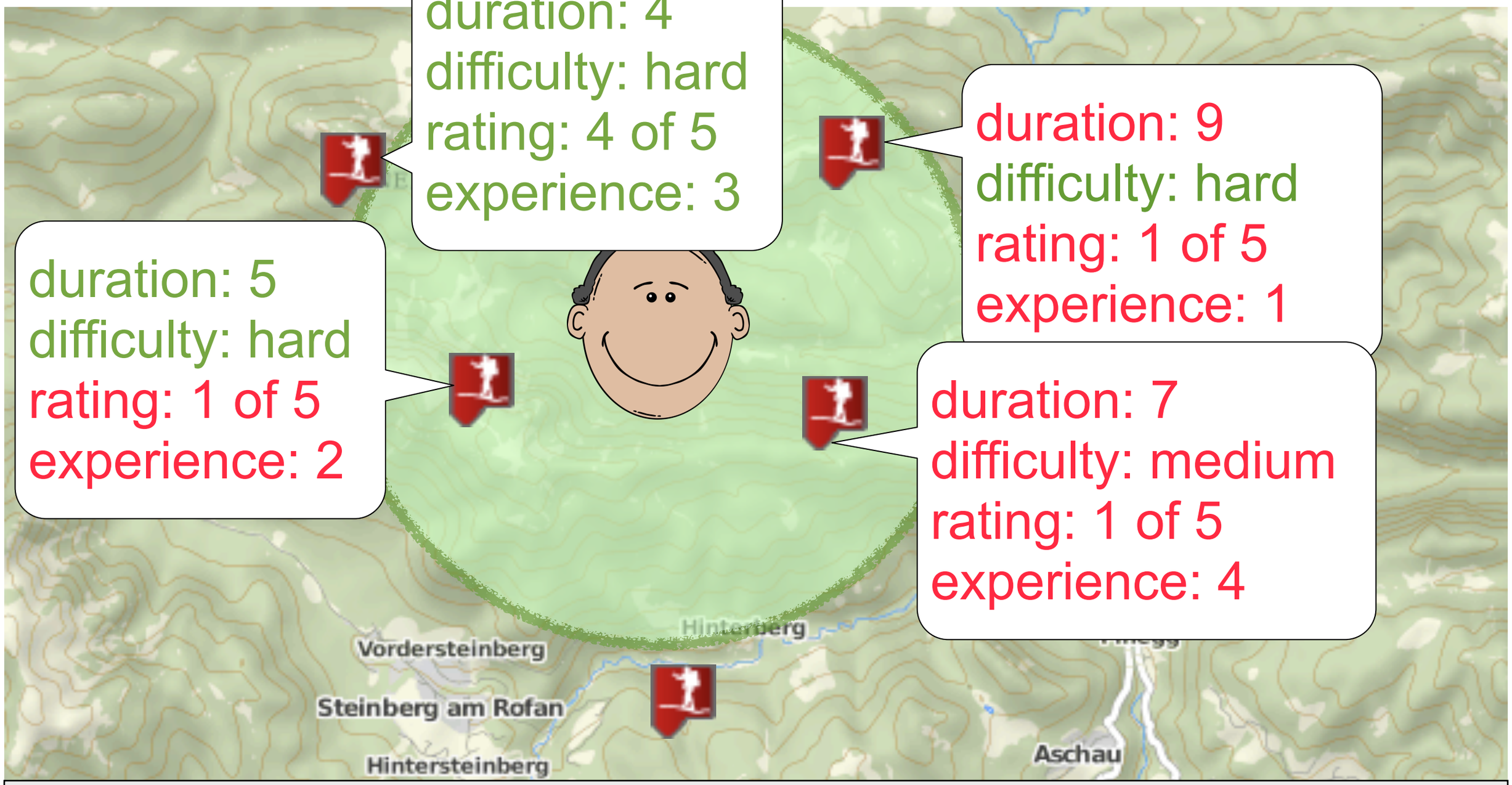
hiking tour: nearby, duration 4 to 6 hours, difficulty *hard*, highest rating, experience level around 3.

Running Example



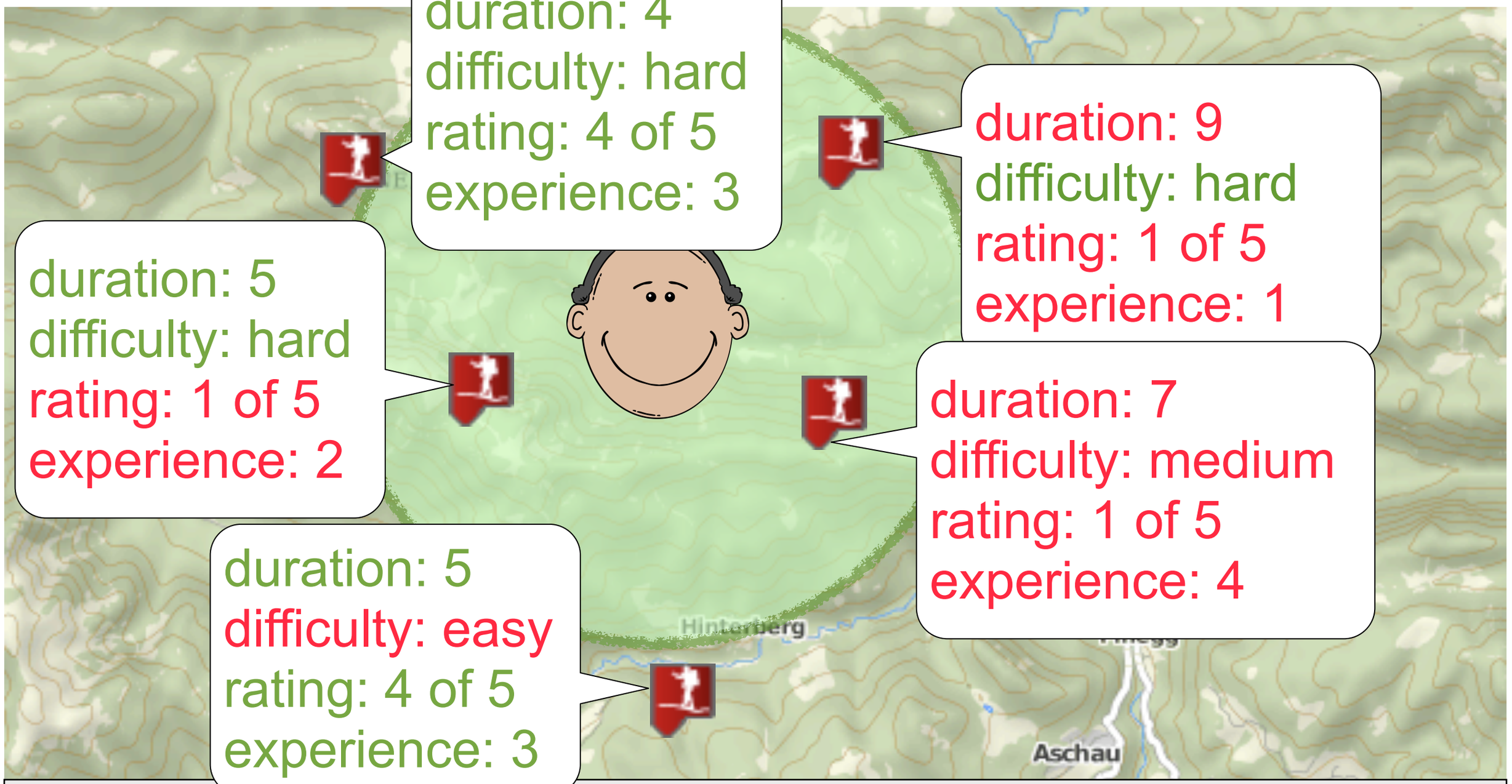
hiking tour: nearby, duration 4 to 6 hours, difficulty *hard*, highest rating, experience level around 3.

Running Example



hiking tour: nearby, duration 4 to 6 hours, difficulty *hard*, highest rating, experience level around 3.

Running Example



hiking tour: nearby, duration 4 to 6 hours, difficulty *hard*, highest rating, experience level around 3.

Derived Requirements

Universität Augsburg University

- The example shows that conventional search
  - often generates too many (flooding) or no results (empty result effect),
  - forces users to iteratively change search parameters,
  - creates long session times from first click to final result.

- What is needed is a query language that
  - avoids empty results and flooding,
  - lets users express wishes concerning basic attributes,
  - lets users state priorities between those basic wishes,
  - delivers results in **one single step**.

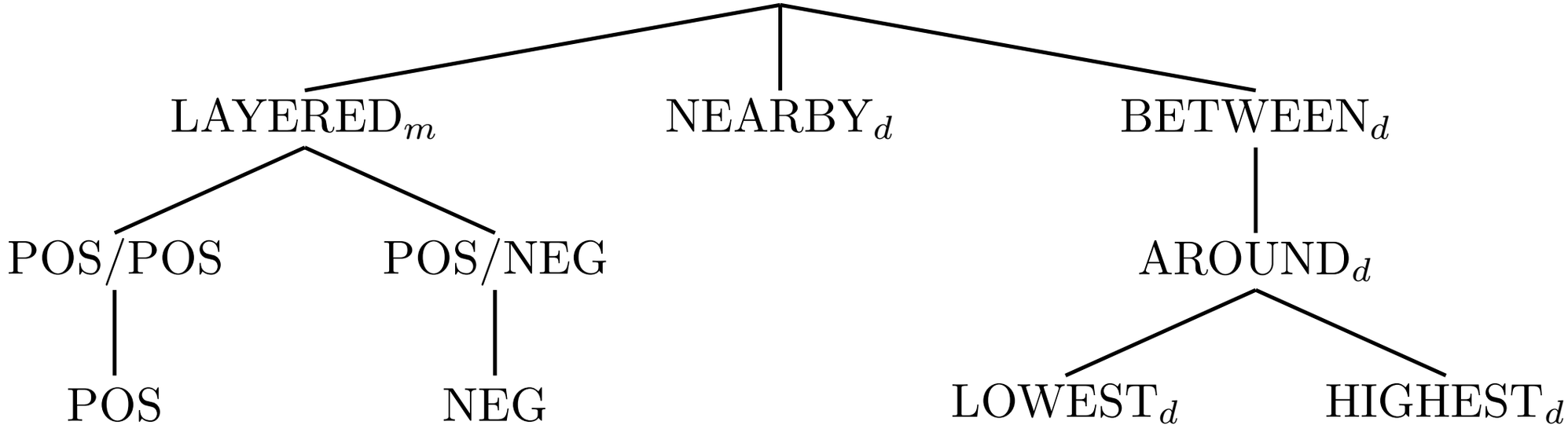
Preference SQL

Universität Augsburg University

- Preference SQL defines a *Preference Algebra* with base and complex preference constructors.
- Based on that algebra, a query language for soft and hard constraints is implemented extending the SQL standard.
- The *Best Matches Only* (BMO) query model returns perfect matches if such matches exist.
- The Query model mitigates flooding and the empty result effect.
- Preference SQL is implemented as Java-middleware that connects to any conventional relational database system.
- The middleware provides its own parser, optimizer and preference algorithms.

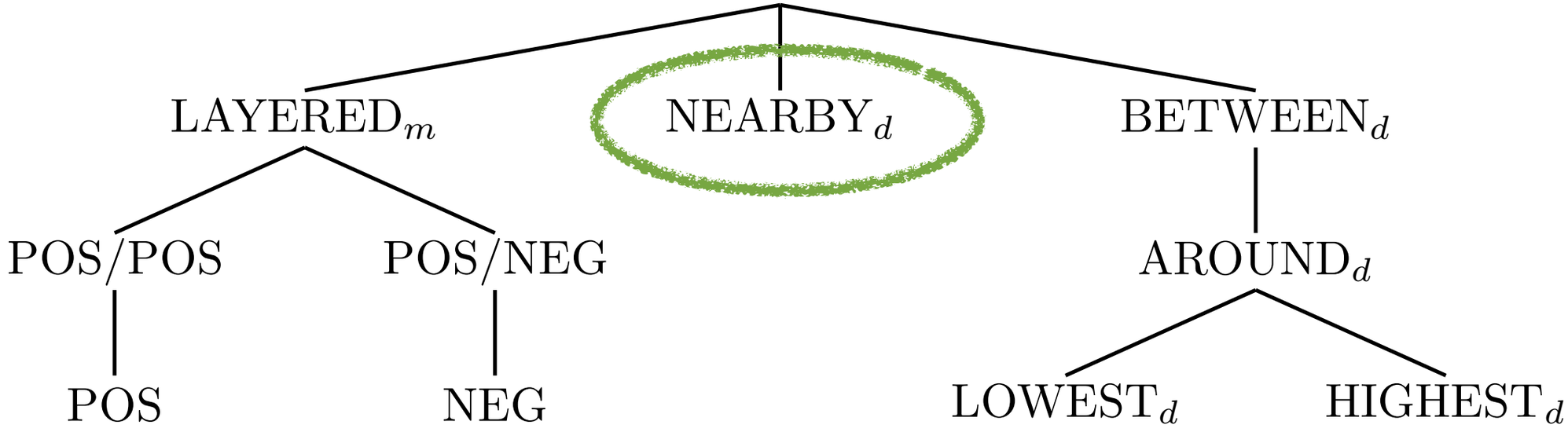
Base Preferences

A hiking tour that is **nearby**, has duration of 4 to 6 hours, difficulty level *hard* and highest rating. If possible, the experience level should be around 3.



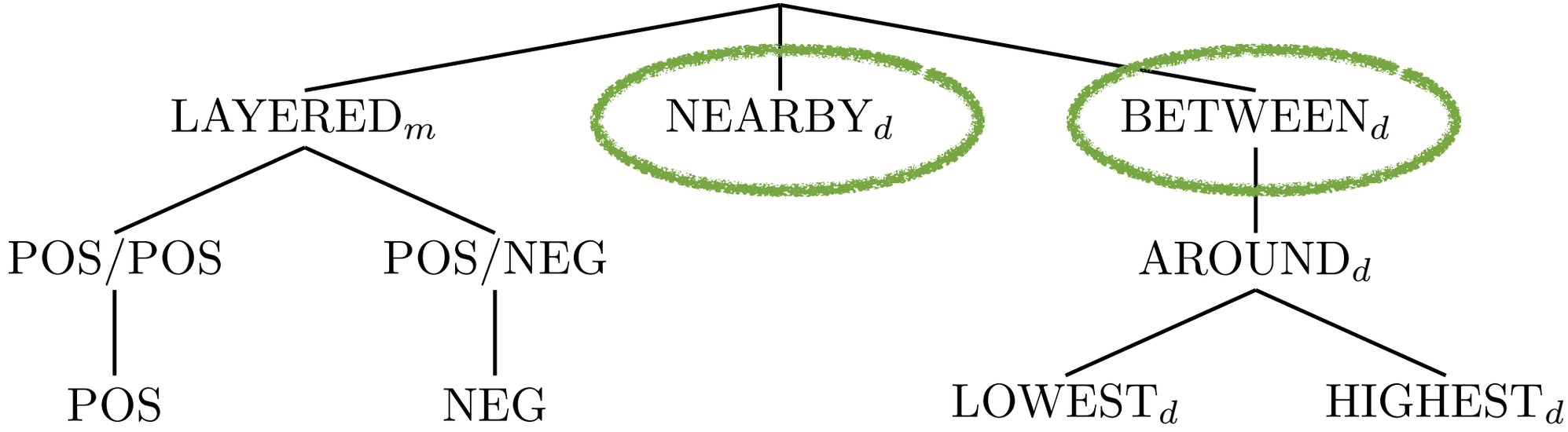
Base Preferences

A hiking tour that is **nearby**, has duration of 4 to 6 hours, difficulty level *hard* and highest rating. If possible, the experience level should be around 3.



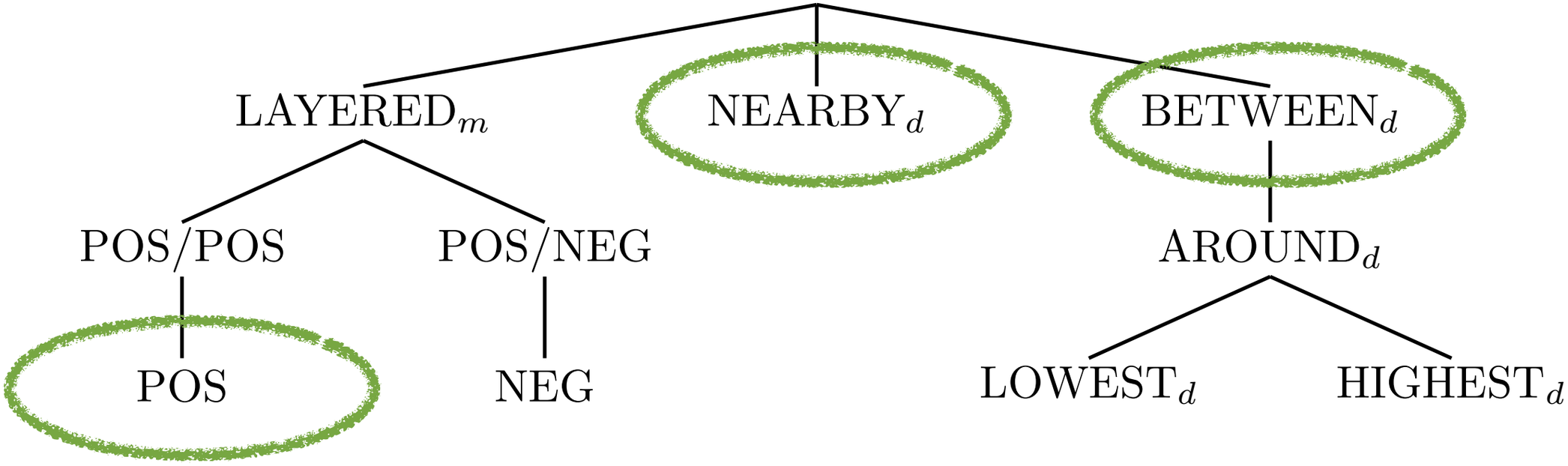
Base Preferences

A hiking tour that is **nearby**, has duration of 4 to 6 hours, difficulty level *hard* and highest rating. If possible, the experience level should be around 3.



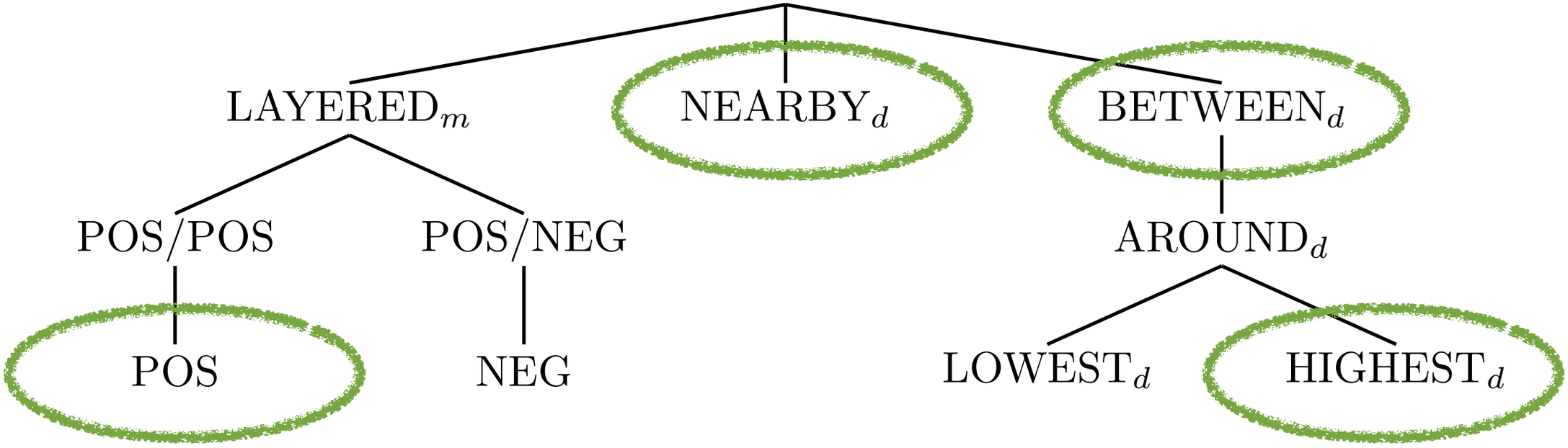
Base Preferences

A hiking tour that is **nearby**, has duration of 4 to 6 hours, difficulty level *hard* and highest rating. If possible, the experience level should be around 3.



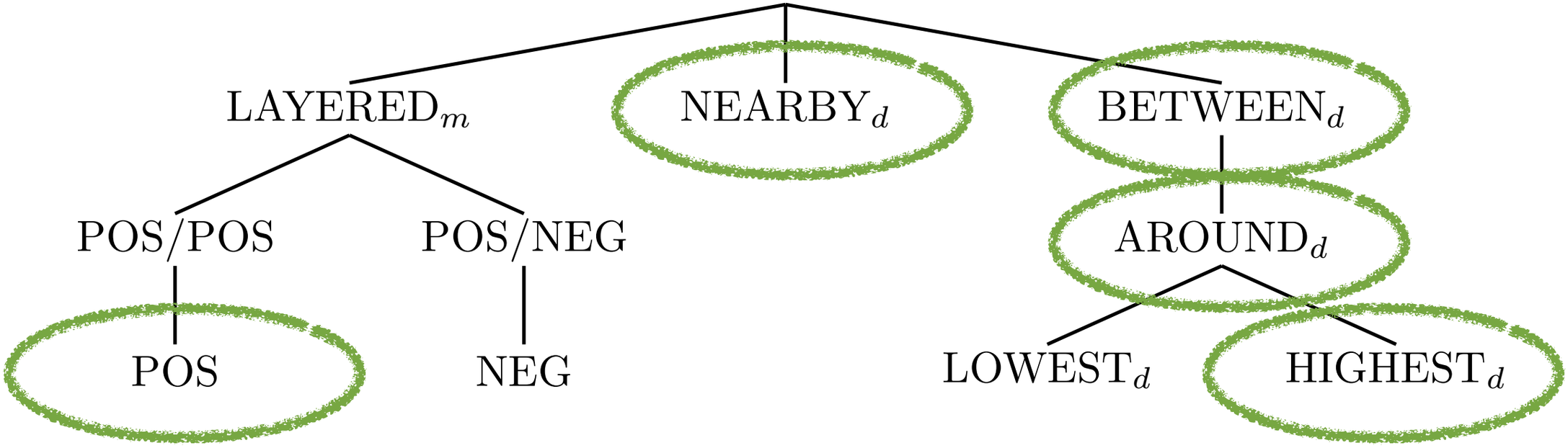
Base Preferences

A hiking tour that is **nearby**, has duration of 4 to 6 hours, difficulty level *hard* and highest rating. If possible, the experience level should be around 3.



Base Preferences

A hiking tour that is **nearby**, has duration of 4 to 6 hours, difficulty level *hard* and highest rating. If possible, the experience level should be around 3.

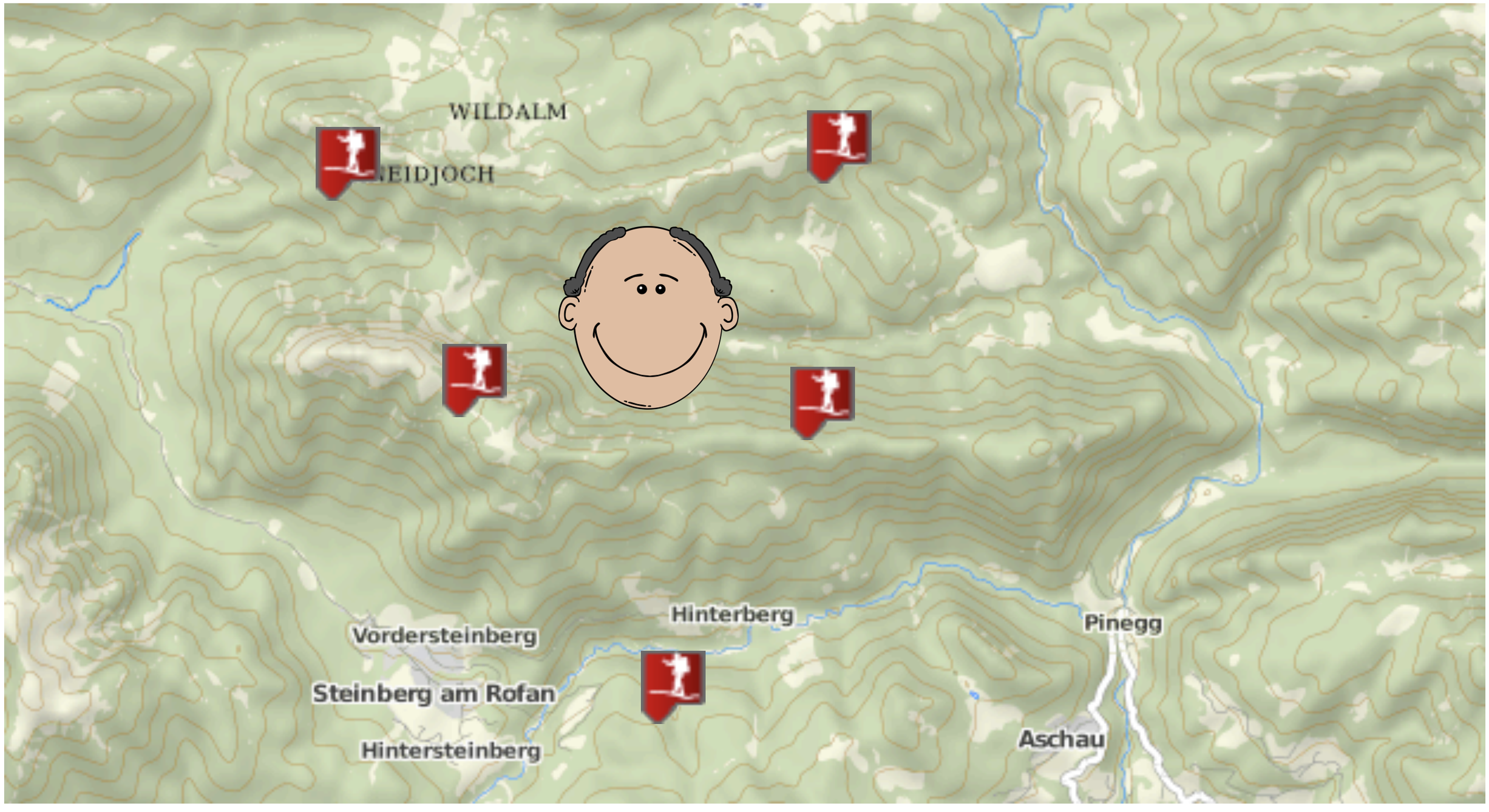


Complex Preferences

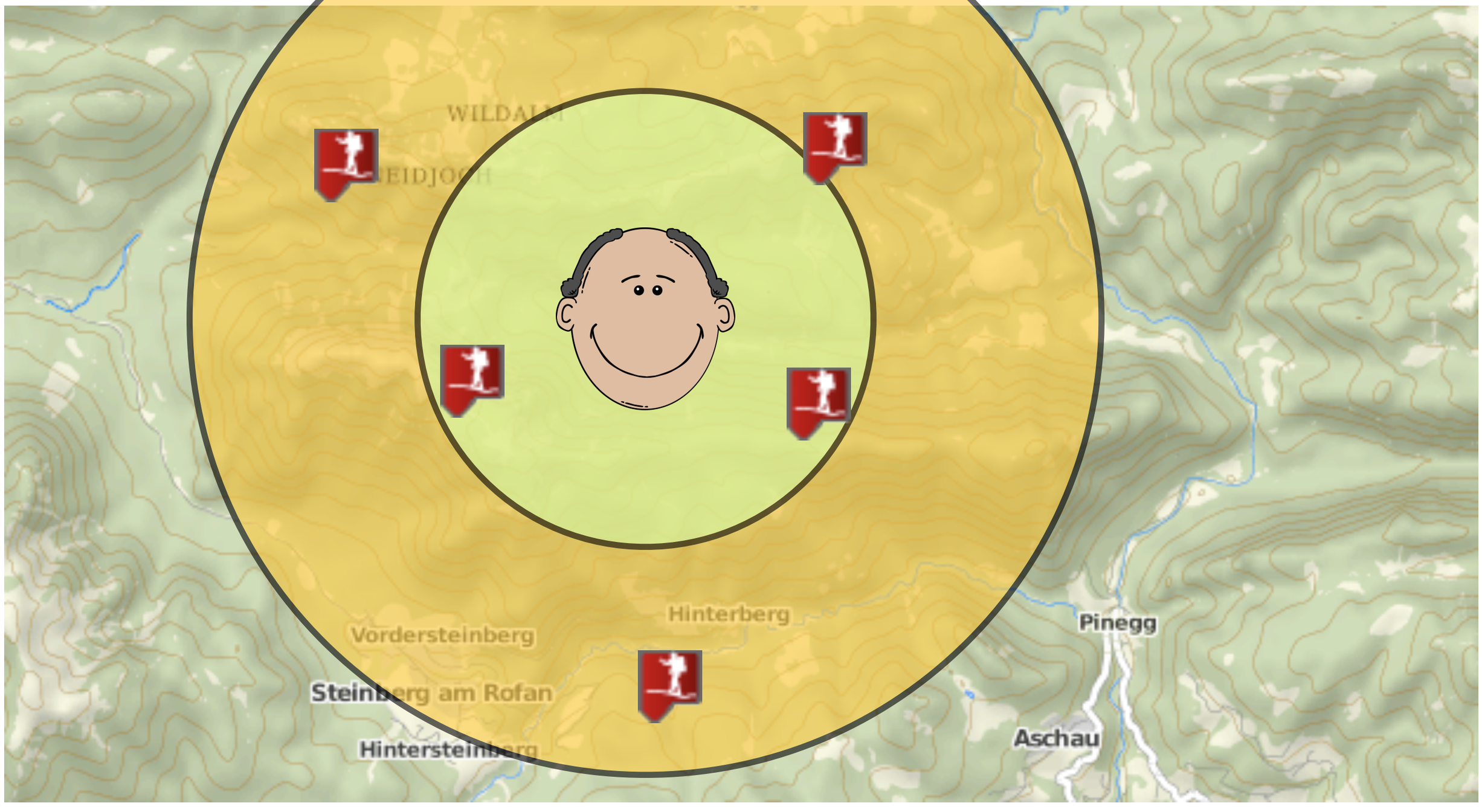
- **Pareto** constructor:
  - Involved preferences are of **equal** importance
  - example: Paul‘s wish for *hard* tours and highest rating

- **Prioritization** constructor:
  - involved preferences are of **ordered** importance
  - example: experience level as Paul‘s tiebreaker preference

NEARBY Preference



NEARBY Preference



Spatial Preferences

- NEARBY is a first representative of *Spatial Preferences*.
- The implementation uses the basic functionality of the underlying PostgreSQL + PostGIS database.
- Several distance functions can be used: ST_Distance, ST_maxDistance

- Performance benchmark:
  - 24000 entries in about 5,5 sec
  - 1000 entries in less than 1 sec
  - largest touristic region in a commercial test case with 1300 entries

Preference SQL Queries

```
SELECT * FROM tours
WHERE region = 'Alpine'
 PREFERRING (
     route NEARBY 47.5494, 11.827211, 2000, ST_maxDistance
     AND duration BETWEEN 4, 6 , 1
     AND difficulty IN ('hard')
     AND rating HIGHEST 5, 1)
     PRIOR TO experience AROUND 3, 1
```

Preference SQL Queries

Universität Augsburg University

base preferences

```
SELECT * FROM tours
WHERE region = 'Alpine'
 PREFERRING (
     route NEARBY 47.5494, 11.827211, 2000, ST_maxDistance
     AND duration BETWEEN 4, 6 , 1
     AND difficulty IN ('hard')
     AND rating HIGHEST 5, 1)
     PRIOR TO experience AROUND 3, 1
```

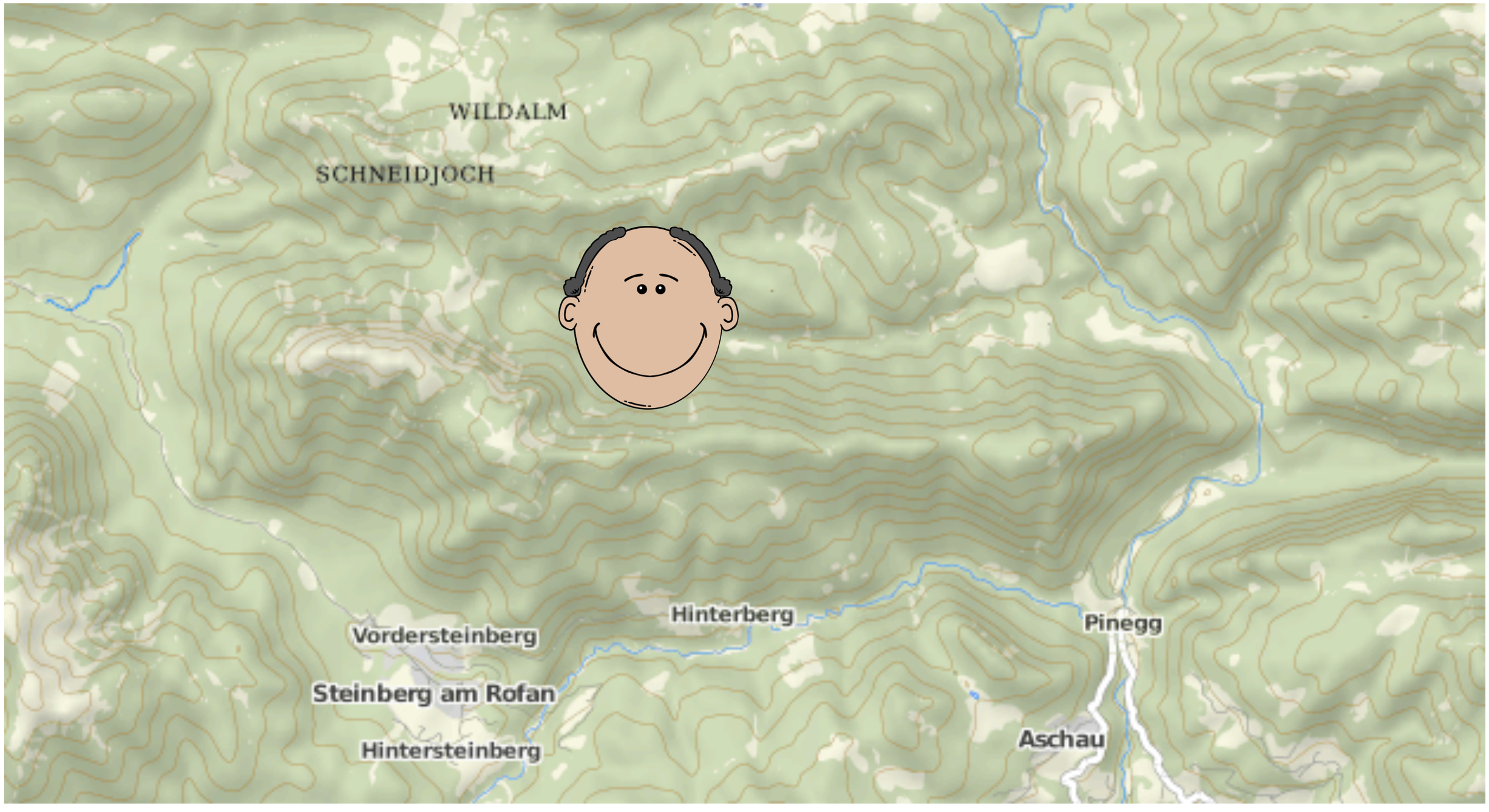
Preference SQL Queries

Universität Augsburg University

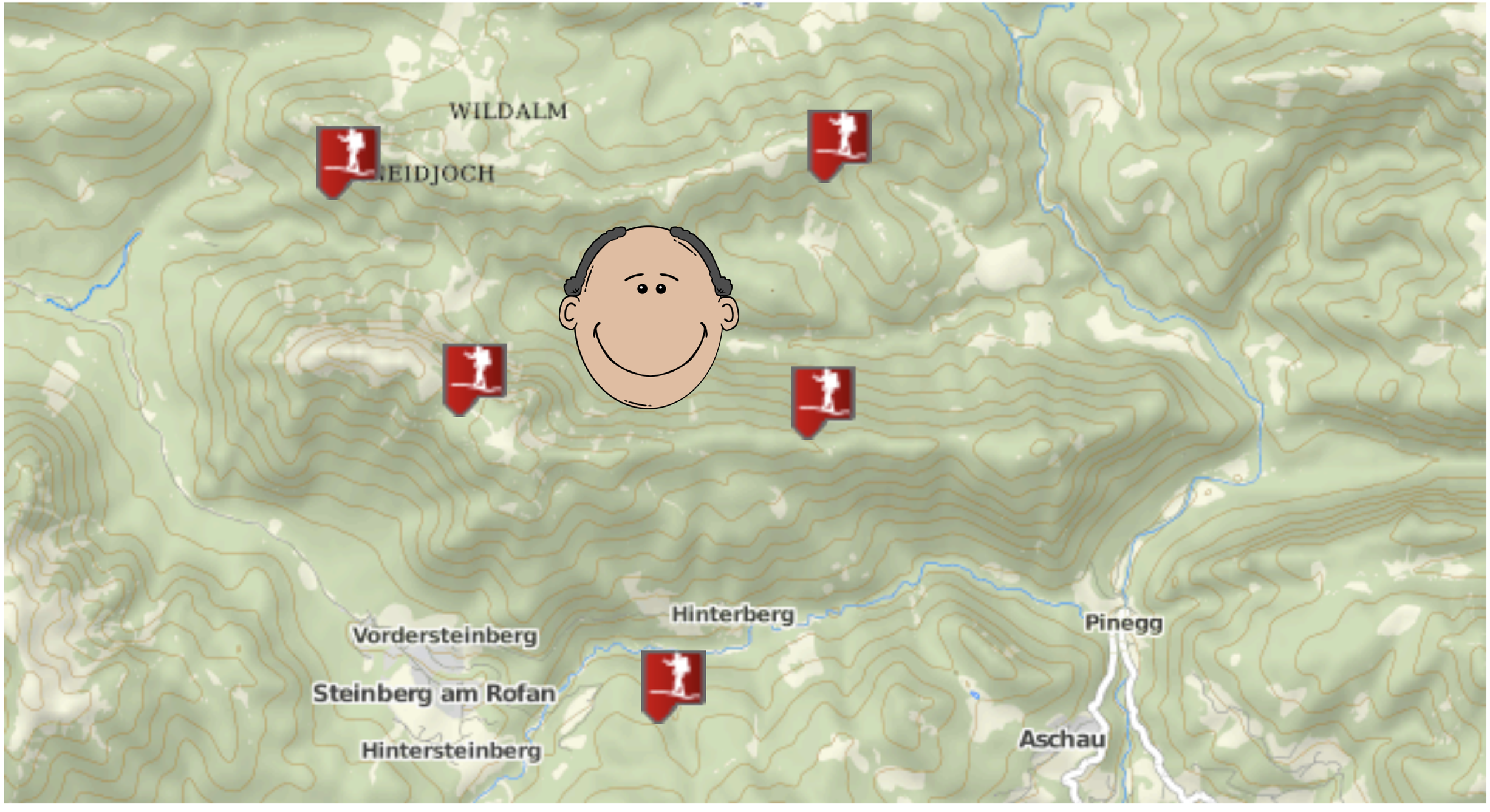
complex preferences

```
SELECT * FROM tours
WHERE region = 'Alpine'
 PREFERRING (
     route NEARBY 47.5494, 11.827211, 2000, ST_maxDistance
     AND duration BETWEEN 4, 6 , 1
     AND difficulty IN ('hard')
     AND rating HIGHEST 5, 1)
     PRIOR TO experience AROUND 3, 1
```

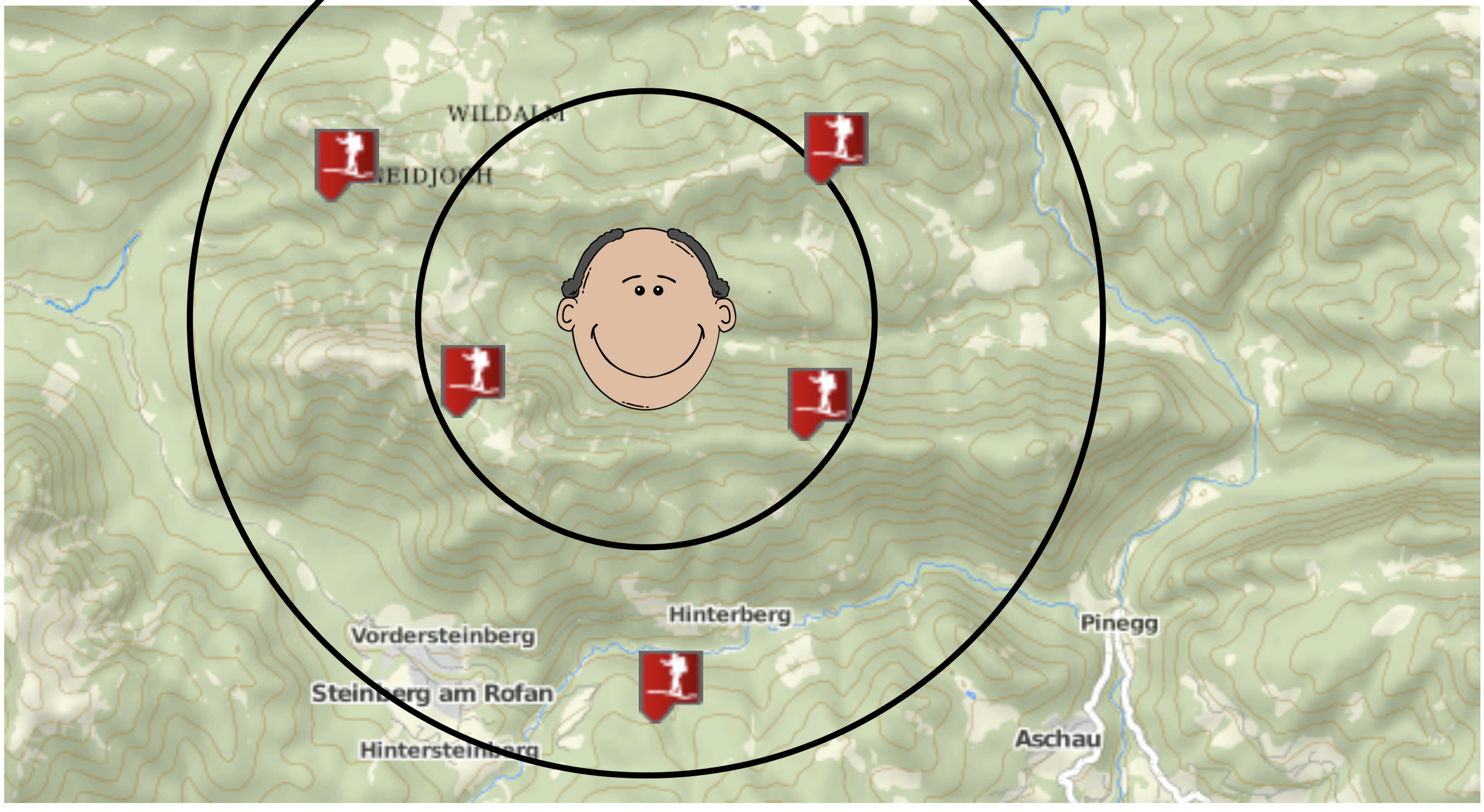
Query Evaluation



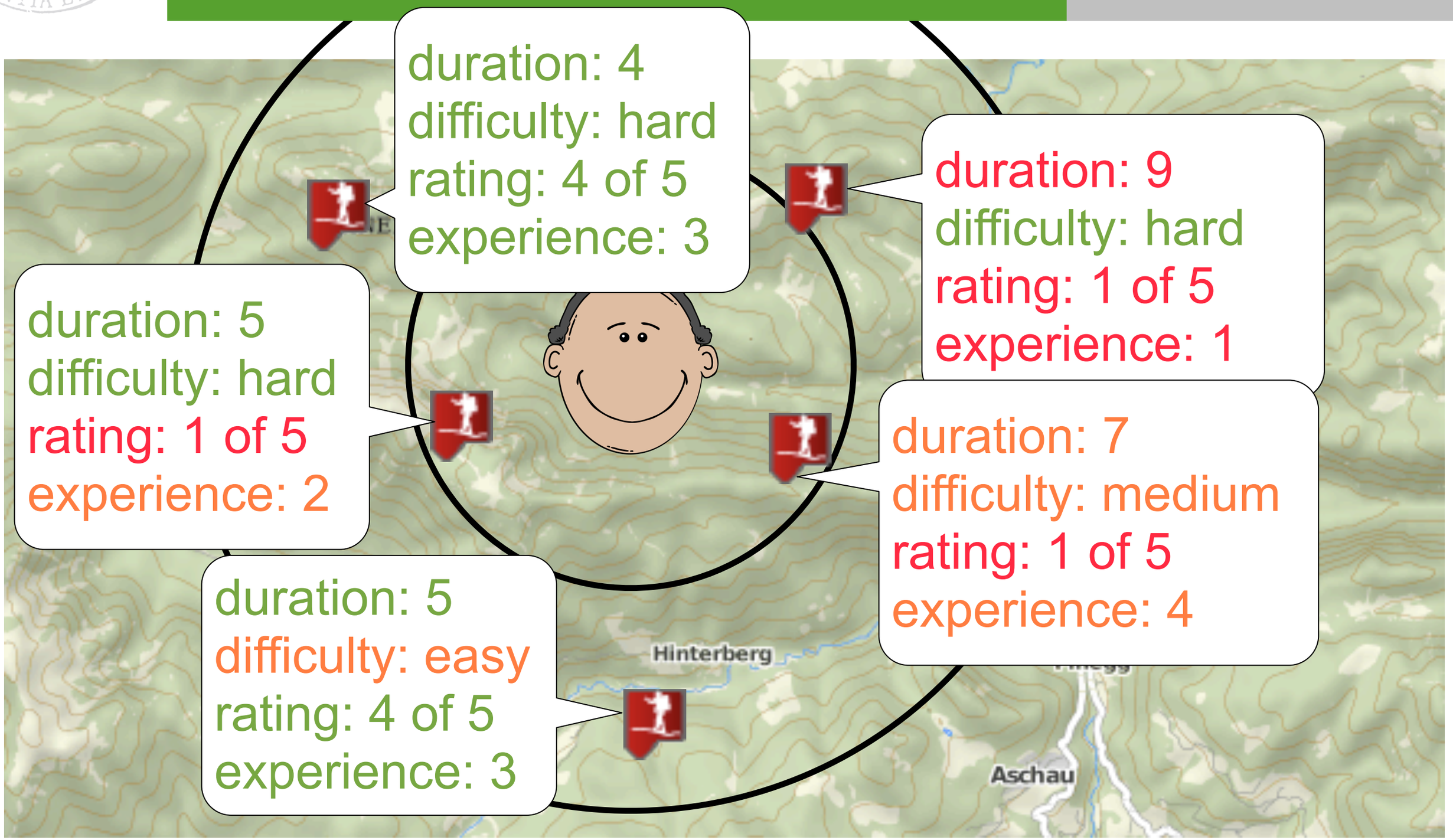
Query Evaluation



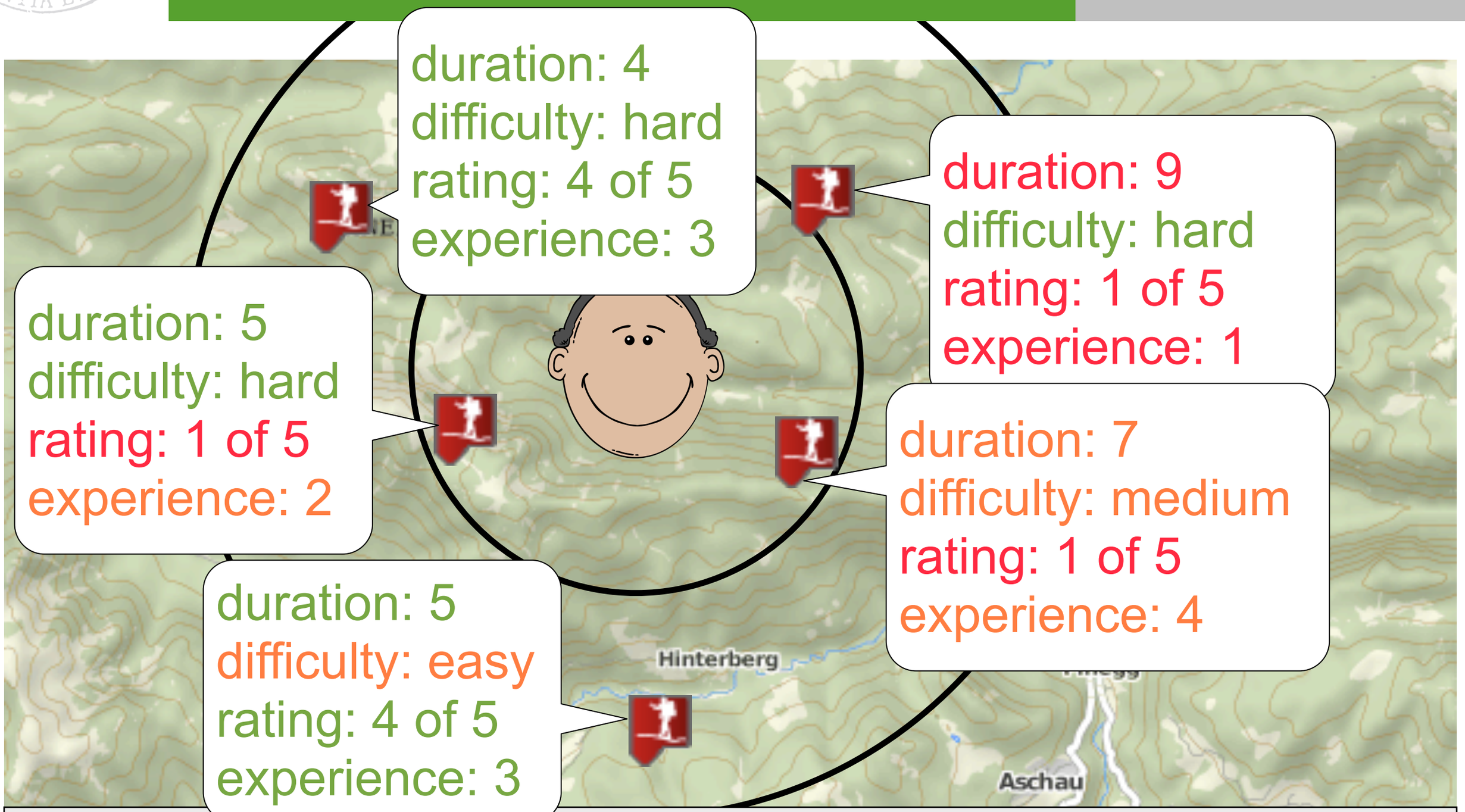
Query Evaluation



Query Evaluation

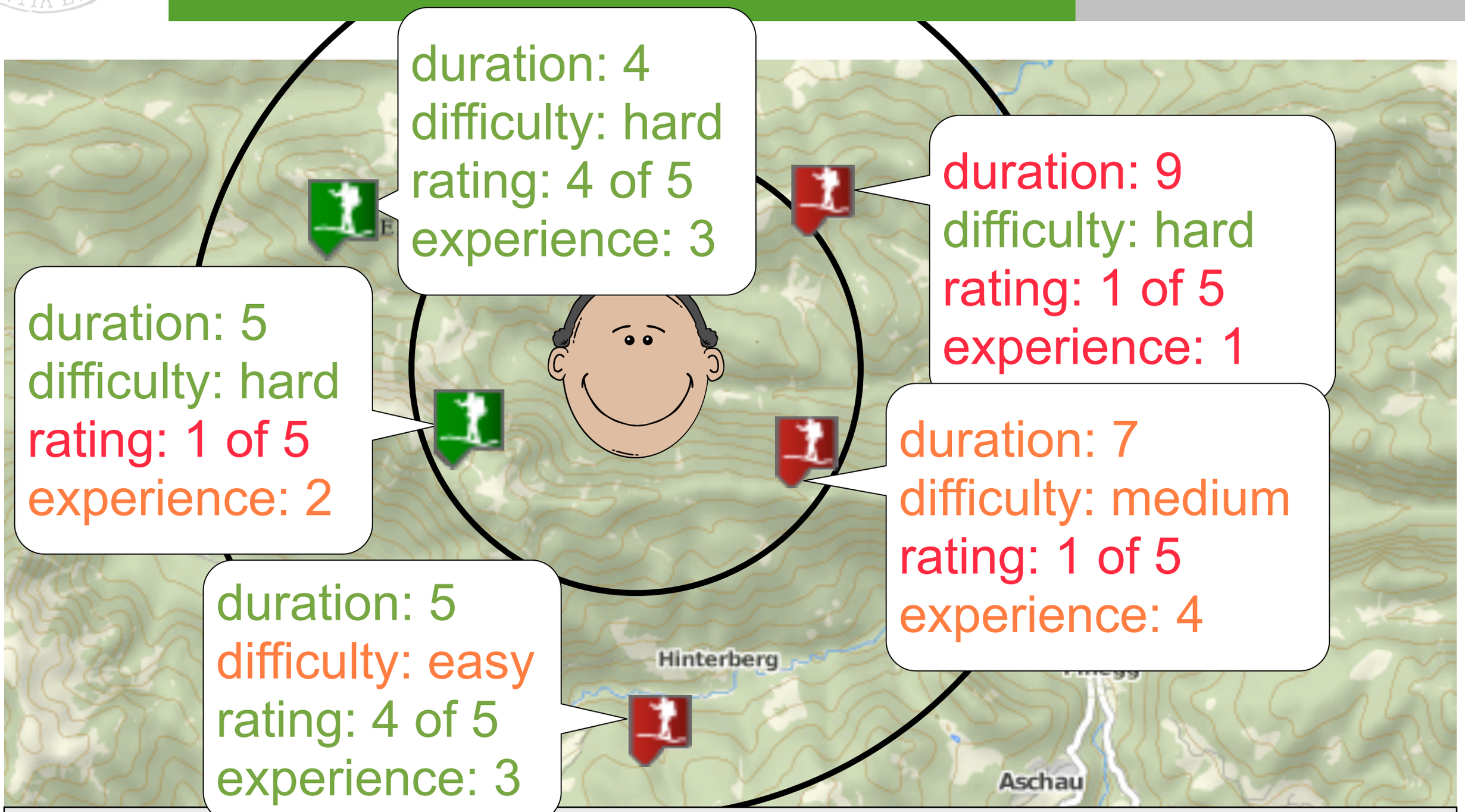


Query Evaluation



hiking tour: nearby, duration 4 to 6 hours, difficulty *hard*, highest rating, experience level around 3.

Query Evaluation



hiking tour: nearby, duration 4 to 6 hours, difficulty *hard*, highest rating, experience level around 3.

Conclusion & Future Work

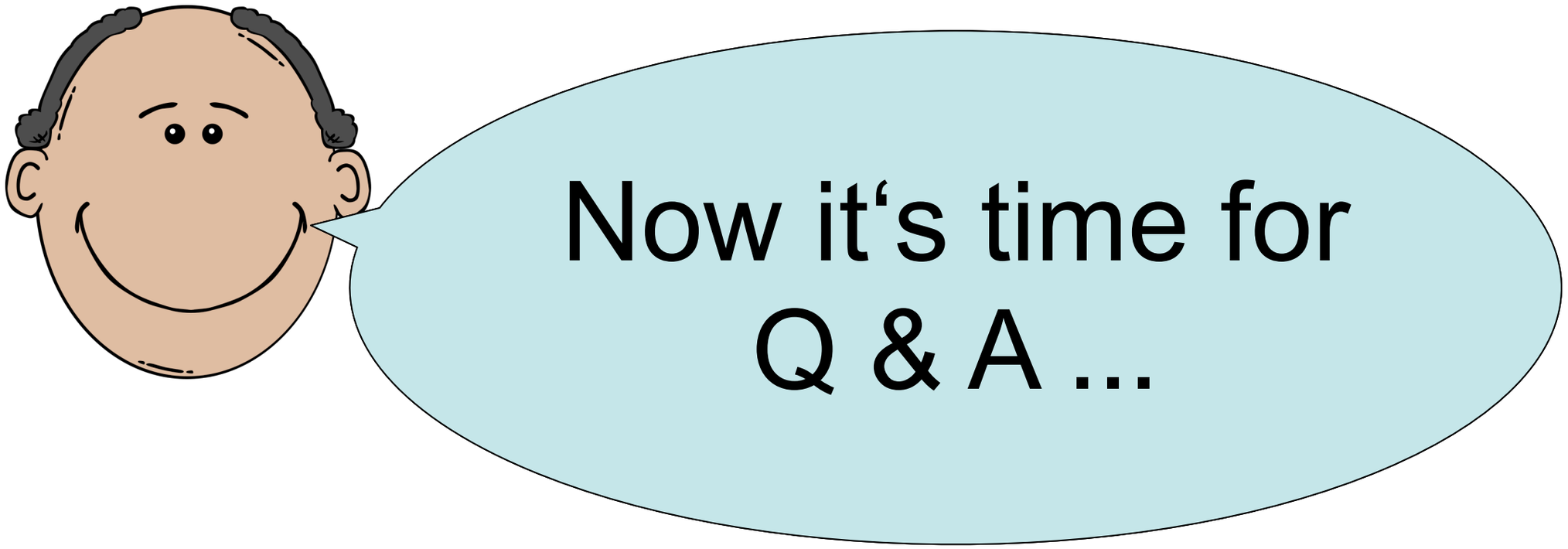
Universität Augsburg University

Conclusion:

- Preference SQL is a first choice for a new generation of LBS.
- It provides One-Step-Search functionality.
- It offers the integration of context models and facilitates user modeling.

Future Work:

- Extension of the SQL standard to support spatial extensions such as SQL-MM Spatial
- Development of new *Spatial Preferences* derived from authentic user scenarios
- *Spatial Group Preferences* for complex planning applications



Spatial Preferences for One-Step-Search in Location-Based Services

F. Wenzel, S. Paulus, W. Kießling

Vienna, 21st Nov. 2011

Showcase

Showcase

- Location-based Android client operating on OpenStreetMap and commercial data
- Application assists users in finding personalized hiking tours, POIs, restaurants, gas stations and ATMs

- The showcase illustrates the main benefits of Preference SQL for LBS:
  - **One-Step-Search** that delivers results without repeated calibration of search parameters
  - **Spatial Preferences** as a new way to incorporate spatial information into search queries
  - **Best personalized results** without the occurrence of flooding or the empty result effect

Schedule: today at **14:40** and **16:25 during coffee break**

For further information, tutorials, downloads and feedback please visit

http://trial.preferencesql.com